**API REST MICRA MAX**

Especificaciones

29 de Junio del 2023  
Alex Rubio Cazador

1. **Introducción:**

**Propósito de la API:**  
El propósito de esta API es proporcionar una interfaz de programación de aplicaciones (API) REST para facilitar la comunicación y el intercambio de datos entre diferentes sistemas. Esta API ha sido desarrollada con el objetivo de permitir a los desarrolladores acceder y manipular ciertos recursos y funcionalidades específicas de nuestro sistema o aplicación de manera eficiente y segura.

Con el uso de esta API, los desarrolladores podrán integrar y utilizar nuestras funcionalidades en sus propias aplicaciones, aprovechando así la capacidad y los datos de nuestro sistema de una manera flexible y personalizada. Además, al seguir los principios de diseño REST, nuestra API se adhiere a estándares ampliamente aceptados, lo que facilita su adopción e implementación en diversos entornos y plataformas.

Al proporcionar una API REST, nuestro objetivo es fomentar la creación de aplicaciones y servicios que puedan interactuar de forma efectiva con nuestro sistema. Esto abre nuevas oportunidades para la colaboración, la innovación y la creación de soluciones que mejoren la experiencia del usuario y brinden un mayor valor agregado a través de la integración de nuestras funcionalidades.

A lo largo de esta documentación, encontrarás información detallada sobre los endpoints disponibles, los parámetros de solicitud, los formatos de respuesta, la autenticación necesaria y las mejores prácticas recomendadas para utilizar nuestra API de manera efectiva. Esperamos que esta documentación sea una guía útil para que los desarrolladores puedan aprovechar al máximo las capacidades de nuestra API y construir aplicaciones poderosas e innovadoras.

**Objetivos y casos de uso**

Proporcionar a los desarrolladores una interfaz estandarizada para acceder y utilizar los datos generados por las sondas y sensores conectados a Micra MAX.

Facilitar la integración de Micra MAX en aplicaciones y sistemas existentes, permitiendo el intercambio de información en tiempo real.

Mejorar la eficiencia de visualización en la captura, análisis y visualización de los datos recopilados por Micra MAX.

Impulsar la innovación al permitir a los desarrolladores crear soluciones personalizadas basadas en la información proporcionada por Micra MAX.

**Casos de Uso**

Monitorización y Control de Procesos: La API permite a los desarrolladores obtener datos en tiempo real de las sondas y sensores conectados a Micra MAX, lo que facilita la monitorización y control de procesos industriales, como la temperatura, humedad, presión, entre otros parámetros críticos.

Integración en Plataformas de Gestión de Activos: Los datos provenientes de Micra MAX pueden ser utilizados en plataformas de gestión de activos, permitiendo a los usuarios monitorear y administrar equipos y sistemas en tiempo real. Por ejemplo, la API puede ser utilizada para recibir alertas automáticas en caso de que algún sensor conectado a Micra MAX indique un problema o un valor fuera de los límites establecidos.

Análisis de Datos y Generación de Informes: La API facilita la extracción de datos almacenados en Micra MAX, lo que permite a los desarrolladores realizar análisis avanzados y generar informes personalizados. Por ejemplo, los datos de los sensores pueden ser utilizados para identificar patrones, tendencias y anomalías, lo que puede ser útil en el mantenimiento predictivo de equipos o en la optimización de procesos.

Estos son solo algunos ejemplos de los objetivos y casos de uso que la API para la conexión con Micra MAX puede abordar. Al proporcionar una interfaz fácil de usar y flexible, nuestra API permite a los desarrolladores aprovechar al máximo los datos generados por Micra MAX, abriendo un mundo de posibilidades para la creación de soluciones personalizadas y la mejora de la eficiencia en diversos sectores y aplicaciones.

**Requisitos previos:**

**Autorización:**

La autorización para acceder a la API se realiza mediante un Token, el cual debe ser incluido en la cabecera de la solicitud con el nombre "X-DTpanel".

El Token o contraseña necesaria para la autorización está definido dentro del equipo Micra MAX.

Es importante destacar que el usuario tiene la capacidad de modificar este Token, lo que brinda flexibilidad y control sobre la seguridad de las solicitudes realizadas al hardware.

Para garantizar una comunicación efectiva y segura, se recomienda generar y utilizar un Token fuerte y único.

Comunicación:

La comunicación con la API de Micra MAX se realiza a través de objetos JSON (JavaScript Object Notation).

Tanto las solicitudes enviadas como las respuestas recibidas deben estar en formato JSON + UTF-8 para garantizar la compatibilidad y facilidad de procesamiento de los datos.

**Métodos de Solicitud:**

Los principales comandos utilizados en la API de Micra MAX son:

* + GET: Utilizado para obtener información o datos específicos de las sondas y sensores conectados a Micra MAX.
  + POST: Utilizado para enviar información o realizar acciones específicas en el hardware de Micra MAX.
  + DELETE: Utilizado para eliminar o desactivar recursos o configuraciones en Micra MAX.

Estos requisitos previos son fundamentales para asegurar una comunicación adecuada con la API de Micra MAX. Asegúrese de cumplir con la autorización mediante el uso del Token proporcionado por el equipo, y de utilizar el formato JSON para las solicitudes y respuestas. Además, familiarícese con los comandos principales (GET, POST y DELETE) para interactuar de manera efectiva con el hardware y obtener los datos necesarios.

**Guía de inicio rápido:**

Ejemplo con CURL:  
curl -vvv -H 'X-DTpanel: **token**' "192.168.1.208/v1/hello.json"

curl -X GET "192.168.1.208/v1/hello.json" -H "X-DTpanel: **token**"

Ejemplos de solicitudes y respuestas  
**RESPONSE BODY:** *200*

[ { "id" : 1, "status" : "Hello World” } ]

**Códigos de Error o Response Body**  
200: Éxito  
201: Éxito Creación.  
400: Petición invalida  
401: No autorizado  
404: Recurso no encontrado  
  
**Endpoints:**

**Enviar una configuración al dispositivo (POST)**  
**/v1/post\_config**  
curl -X POST "IP/v1/config"   
 -H "X-DTpanel: token"   
 -H "Content-type: application/json"   
 -d @request\_body  
  
*donde @request\_body = json con la configuración***RESPONSE BODY:** *201*

**Recibir una configuración al dispositivo (GET)**  
**/v1/get\_config**  
curl -X GET "IP/v1/config"   
 -H "X-DTpanel: token"   
  
**RESPONSE BODY:** *200*{

"inputType": {

"processInputType": null,

"loadCellInputType": {

"loadCellInputTypeValues": "150mv"

},

"thermometerInputType": null

},

"display": {

"optionsOfDisplayConfig": {

"decimalsDisplay": 2,

"linearisationPoints": 11,

"roundingFilter": "01",

"balancedFilter": 1

},

"values": [

{

"inputValue": 0,

"displayValue": 0

},

{

"inputValue": 0,

"displayValue": 0

},

{

"inputValue": 0,

"displayValue": 0

},

{

"inputValue": 0,

"displayValue": 0

},

{

"inputValue": 0,

"displayValue": 0

},

{

"inputValue": 0,

"displayValue": 0

},

{

"inputValue": 0,

"displayValue": 0

},

{

"inputValue": 0,

"displayValue": 0

},

{

"inputValue": 0,

"displayValue": 0

},

{

"inputValue": 0,

"displayValue": 0

},

{

"inputValue": 0,

"displayValue": 0

}

],

"displayInputError": "off",

"displayEco": {

"ecoState": "off",

"ecoValue": 0

},

"brightType": "high",

"tareValue": 1,

"runModeColor": "green",

"progModeColor": "orange"

},

"setPoints": [

{

"onOffType": "off",

"value": 0,

"comparisonValue": "net",

"actionMode": "high",

"noNcType": "no",

"colorAlarmType": "noChange",

"activationType": "delay",

"activationValue": 0

},

{

"onOffType": "off",

"value": 0,

"comparisonValue": "net",

"actionMode": "high",

"noNcType": "no",

"colorAlarmType": "noChange",

"activationType": "delay",

"activationValue": 0

},

{

"onOffType": "off",

"value": 0,

"comparisonValue": "net",

"actionMode": "high",

"noNcType": "no",

"colorAlarmType": "noChange",

"activationType": "delay",

"activationValue": 0

},

{

"onOffType": "off",

"value": 0,

"comparisonValue": "net",

"actionMode": "high",

"noNcType": "no",

"colorAlarmType": "noChange",

"activationType": "delay",

"activationValue": 0

}

],

"rssConfig": {

"bauds": 9600,

"id": 1,

"protocol": "ascii",

"replyDelay": "no\_delay"

},

"ipAddress": null,

"analogueConfig": {

"displayHigh": 0,

"displayLow": 0,

"overrange": "high"

},

"logicFunctionConfig": {

"pin2": 1,

"pin3": 2,

"pin4": 6

},

"programmingLockCodeConfig": {

"code": 0,

"lkSetPoints1": false,

"lkSetPoints2": false,

"lkSetPoints3": false,

"lkSetPoints4": false,

"input": false,

"display": false,

"analogueOutput": false,

"rsOutput": false,

"logicFunctions": false,

"tareKey": false,

"directSetpoints": false,

"brightnessColor": false,

"filterP": false,

"maxMin": false

}

}

**Recibir información del display (GET)**  
**/v1/get\_display**  
curl -X GET "IP/v1/get\_display"   
 -H "X-DTpanel: token"   
  
**RESPONSE BODY:** *200*Respuesta el Json con los datos del display de eses momento, maximo, mínimos y setpoints.  
  
**Reset de la tara (POST)**  
**/v1/reset\_tare**  
curl -X POST "IP/v1/reset\_tare"   
 -H "X-DTpanel: token"   
  
**Reset Max (POST)**  
**/v1/reset\_max**  
curl -X POST "IP/v1/reset\_max"   
 -H "X-DTpanel: token"   
  
**Reset Min (POST)**  
**/v1/reset\_min**  
curl -X POST "IP/v1/reset\_min"   
 -H "X-DTpanel: token"

**Tara (POST)**  
**/v1/tare**  
curl -X POST "IP/v1/tare"   
 -H "X-DTpanel: token"

**Factory Reset (POST)**  
**/v1/factory\_reset**  
curl -X POST "IP/v1/fatory\_reset"   
 -H "X-DTpanel: token"   
 **Obtener información del equipo (GET)**  
**/v1/get\_info**  
curl -X GET "IP/v1/get\_info"   
 -H "X-DTpanel: token"   
  
Deberíamos obtener: Model Name, Serial-NUmber, Date Manufactured, Software version, Options installed, así como los parámetros de la telemetría del aparato.