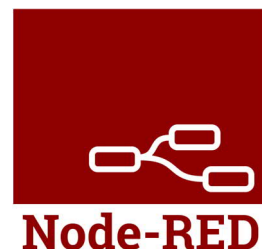


REST API en Node-RED con autenticación



Instrucciones sobre como leer y escribir
variables del PLCnext desde Node-RED
utilizando REST API con autenticación



©PHOENIX CONTACT 2021-10-26

1 Descripción

Este documento describe el procedimiento leer
y escribir variables del PLCnext desde Node-RED
utilizando REST API con autenticación

2 Requisitos de hardware

- AXC 1152
- AXC 2152

3 Requisitos de software

- PLCnext Engineer, versión ≥ 2021.3
- Node-RED, versión $\geq 1.3.5$

Tabla de contenido

1	Descripción.....	1
2	Requisitos de hardware	1
3	Requisitos de software	1
4	Proyecto en PLCnext Engineer	2
4.1	Configuración del HMI Webserver.....	2
4.2	Configuración de acceso a las variables	2
4.3	Configuración de la seguridad.....	3
5	Proyecto en Node-RED.....	3
5.1	Generar el código de autenticación	3
5.2	Generar el token de acceso (Bearer token).....	4
5.3	Leer variables con el método GET	5
5.4	Escribir variables con el método PUT	7



Encontrará más información sobre la implementación de la interfaz de datos REST API en:
https://www.plcnext.help/te/Service_Components/REST_data_interface/REST_data_interface_Introduction.htm

4 Proyecto en PLCnext Engineer

- Abra un proyecto nuevo o existente

4.1 Configuración del HMI Webserver

- El proyecto de PLCnext Engineer debe contener una página HMI (puede estar vacía) para inicializar el componente PLCnext HMI

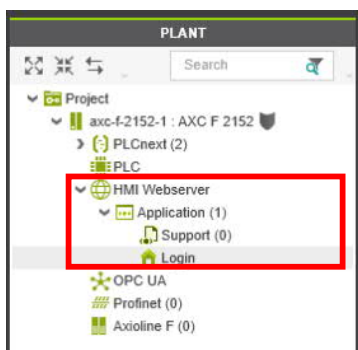


Fig. 1 HMI Webserver

- Despliegue la sección HMI Webserver

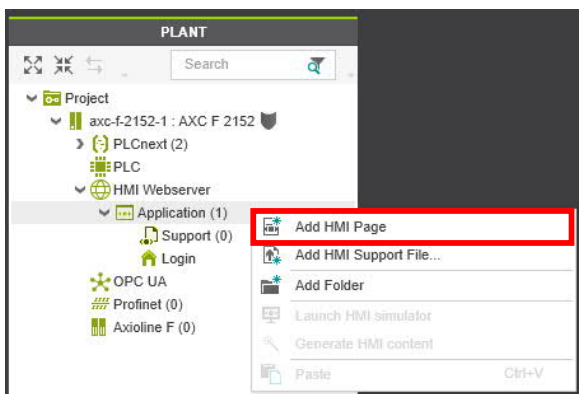


Fig. 2 Agregar página

- Haga clic derecho en "Application" y agregue una nueva página HMI mediante "Add HMI Page"

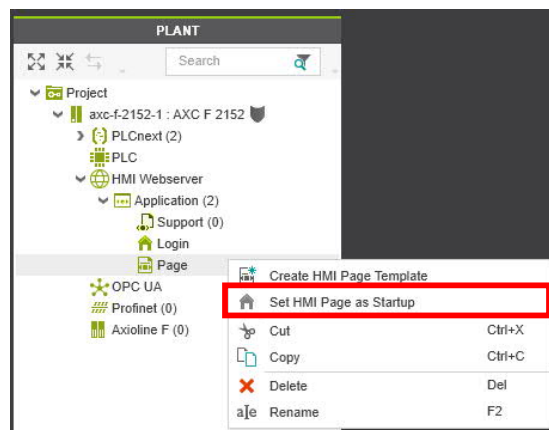


Fig. 3 Página como principal

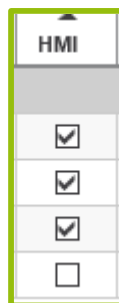
- Haga clic derecho sobre la página y seleccione la opción "Set HMI Page as Startup"

4.2 Configuración de acceso a las variables

- A todas las variables de PLC a las que se desea acceder a través de la interfaz de datos REST se las debe seleccionar en su casillero HMI correspondiente

Name	Type	Usage	Tr...	Comment	Init	Ret...	Co...	OPC	HMI
rTemperatura	REAL	Local	<input type="checkbox"/>		REAL#0 0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
rHumedad	REAL	Local	<input type="checkbox"/>		REAL#0 0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
rContenido	REAL	Local	<input type="checkbox"/>		REAL#0 565	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Enter variable name here									

Fig. 4 Selección de variables



- Dentro de la hoja de variables del programa del proyecto, seleccionar el recuadro "HMI"

4.3 Configuración de la seguridad

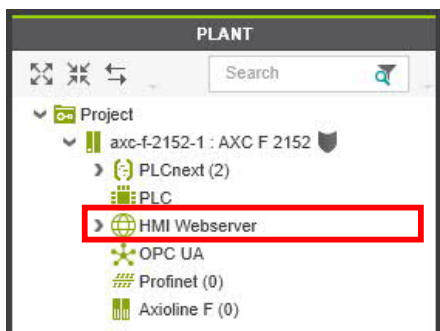


Fig. 5 Abrir HMI Webserver

- Haga doble clic en HMI Webserver

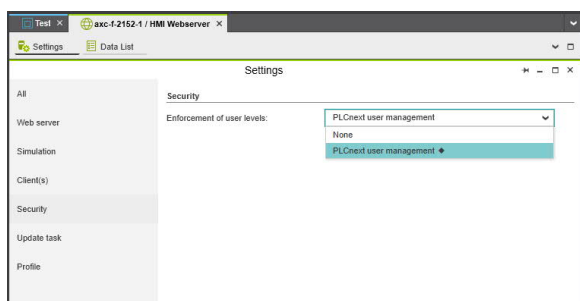


Fig. 6 Seguridad en HMI Webserver

- Dentro de "Security" en "Enforcement of user levels" seleccione la opción "PLCnext user management" (en nuevos proyectos viene activado por default)

5 Proyecto en Node-RED

- Abra un proyecto existente o cree uno nuevo

5.1 Generar el código de autenticación

5.1.1 Flow de ejemplo



Fig. 7 Flow en Node-RED

5.1.2 Configuración del nodo inject

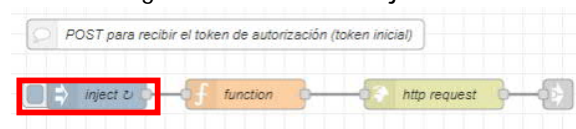


Fig. 8 Nodo inject

- Haga doble clic en el nodo inject

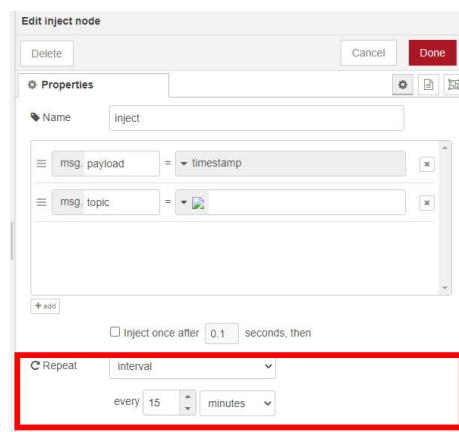


Fig. 9 Configuración de repetición

- Selecione la opción de repetir cada 15 minutos. Esto se debe a que el Bearer token caduca a los 20 minutos por seguridad

5.1.3 Configuración del nodo function

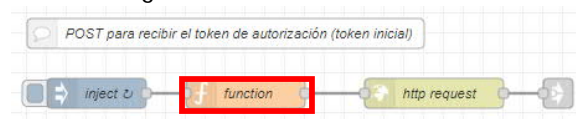


Fig. 10 Nodo function

- Haga doble clic en el nodo function y defina el siguiente mensaje para definir el payload que nos permitirá recibir el token de autenticación

```
1 msg.payload =
2 {
3   "scope": "variables"
4 }
5 return msg;
```

Fig. 11 Mensaje del nodo function

5.1.4 Configuración del nodo http request

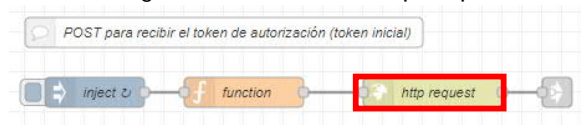


Fig. 12 Nodo http request

- Haga doble clic en el nodo http request

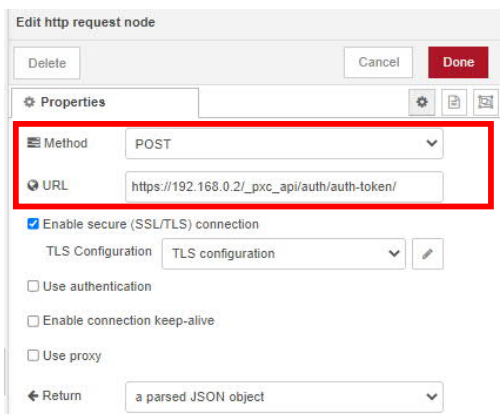


Fig. 13 Config. nodo http request

- Método: POST
- URL:
https://%PlcAddress%/_pxc_api/auth/auth-token/

Donde:

- %PlcAddress%: dirección IP del PLC
- En "Return" seleccionar la opción "a parsed JSON object"

5.2 Generar el token de acceso (Bearer token)

5.2.1 Flow de ejemplo



Fig. 14 Flow en Node-RED

5.2.2 Conexión con el flow anterior

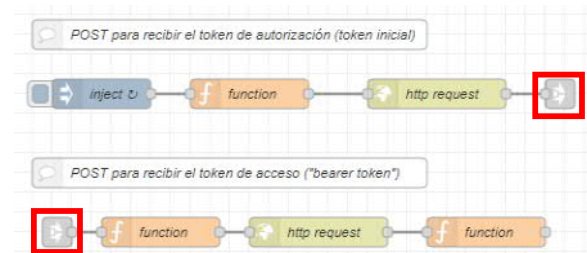


Fig. 15 Conexión con el flow anterior

- Sugerencia: utilice los nodos "Link In" y "Link Out" uniendo el final del flow anterior con el principio del actual.

5.2.3 Configuración del primer nodo function



Fig. 16 Nodo function

- Haga doble clic en el nodo function y defina el siguiente mensaje para seleccionar la propiedad "code" del mensaje anterior y añadirla al nuevo mensaje para obtener el token de acceso

```
1 msg.payload =
2 {
3   "code": msg.payload.code,
4   "grant_type": "authorization_code",
5   "username": "admin",
6   "password": "a78e220b"
7 }
8 return msg;
```

Fig. 17 Mensaje del nodo function

- Es necesario aclarar que en "username" debe ir el usuario del PLC y en "password" la contraseña asociada a dicho usuario. Por default, el PLC trae asociado el usuario "admin" y la contraseña grabada en la carcasa del dispositivo.

5.2.4 Configuración del nodo http request



Fig. 18 Nodo http request

- Haga doble clic en el nodo http request

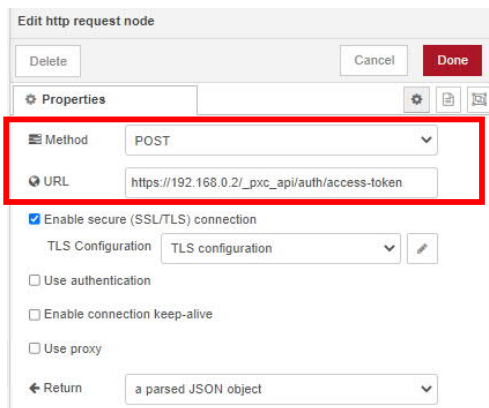


Fig. 19 Config. nodo http request

- Método: POST
- URL:
https://%PlcAddress%/_pxc_api/auth/access-token

Donde:

- %PlcAddress%: dirección IP del PLC
- En "Return" seleccionar la opción "a parsed JSON object"

5.2.5 Configuración del segundo nodo function



Fig. 20 Nodo function

- Haga doble clic en el nodo function y defina el siguiente mensaje, el cual le asignará a una variable global el valor del Bearer token. Al ser una variable global, puede ser utilizada en distintos flows

```
1 //Crear una variable global con el valor del payload anterior
2 //Una variable global puede ser accedida desde cualquier flow
3 var access_token = msg.payload.access_token;
4 global.set("bearer_token", access_token);
5
6 //Leer el valor de la variable global
7 msg.payload = global.get("bearer_token");
8 return msg;
```

Fig. 21 Mensaje del nodo function

- La variable global es "bearer_token" en el ejemplo actual
- El Bearer token es el que debemos utilizar en todos los demás headers de las consultas http.

5.3 Leer variables con el método GET

5.3.1 Flow de ejemplo

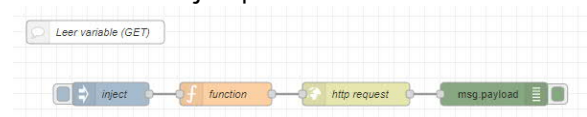


Fig. 22 Flow en Node-RED

5.3.2 Configuración del nodo inject

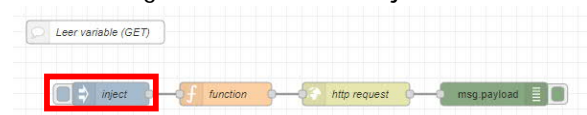


Fig. 23 Nodo inject

- Haga doble clic en el nodo inject

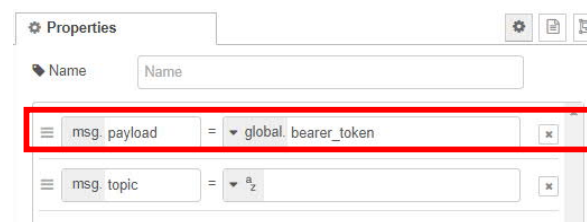


Fig. 24 Asignar la variable global

- Configurar en la sección de msg.payload como:
global.%Nombre_de_la_variable_global%.
- De esta manera, el payload que ingresa al flow corresponde al valor de la variable global, la cual posee el valor del Bearer

token. Este Flow solo se activará cuando se hace clic en el nodo "inject"

5.3.3 Configuración del nodo function



Fig. 25 Nodo function

- Haga doble clic en el nodo function y defina el siguiente mensaje, mediante el cual se le asigna en el encabezado (header) el valor del Bearer token

```
1 = msg.headers = {
2   Authorization: msg.payload
3 = }
4 return msg;
```

Fig. 26 Mensaje del nodo function

5.3.4 Configuración del nodo http request



Fig. 27 Nodo http request

- Haga doble clic en el nodo http request

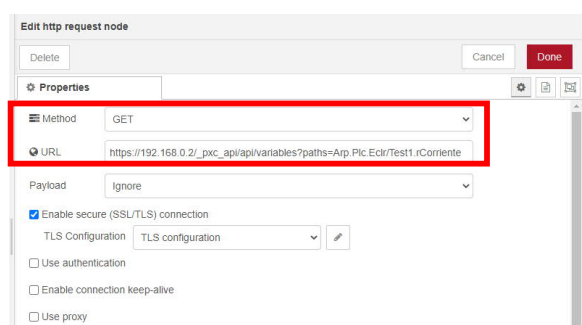


Fig. 28 Config. del nodo http request

- Método: GET
- URL:
https://%PlcAddress%/_pxc_api/api/variables?paths=%VariablePath1%,...

Donde:

- %PlcAddress%: dirección IP del PLC
- %VariablePath1%: ruta relativa o completa de la primera variable (Arp.Plc.Eclr/), así como un índice, índices y / o rango de índices opcionales en el caso de una variable de matriz (PartArray[2], PartArray[2; 4], PartArray[2; 4; 6-8])

- Es posible solicitar la lectura de más de una variable en un mismo request.

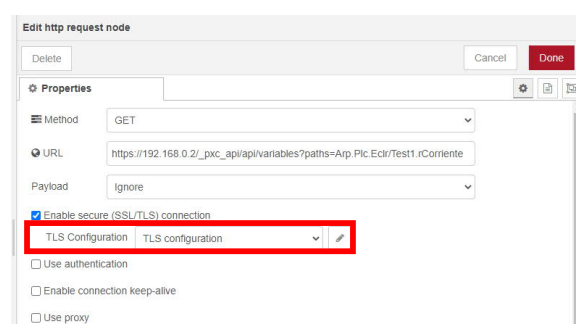



Fig. 29 Certificados

- Seleccione  y deshabilite los certificados y la opción "Verify server certificate".

5.3.5 Mensaje de respuesta

- La respuesta es el código de estado HTTP 200(OK) para una lectura exitosa de la variable junto con los datos JSON resultantes; de lo contrario se lee el código de estado HTTP correspondiente (por ejemplo, 400= Solicitud incorrecta, 408= Tiempo de espera de solicitud, 500= Error interno del servidor).

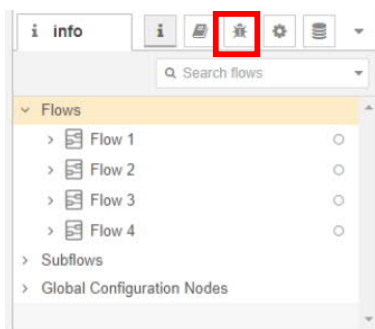


Fig. 30 Modo Debug en Node-RED

- Seleccionar la opción "Debug messages"

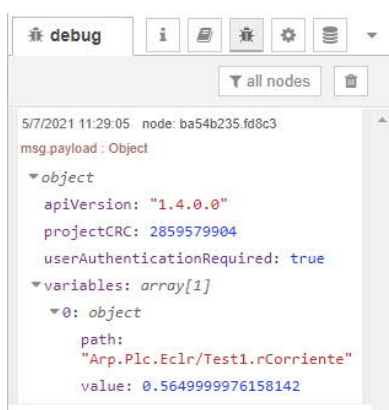


Fig. 31 Respuesta JSON

- Respuesta recibida que contiene la lectura de la variable

5.4 Escribir variables con el método PUT

5.4.1 Flow de ejemplo

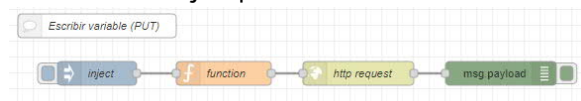


Fig. 32 Flow en Node-RED

5.4.2 Configuración del nodo inject

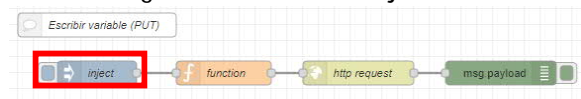


Fig. 33 Nodo inject

- Haga doble clic en el nodo inject

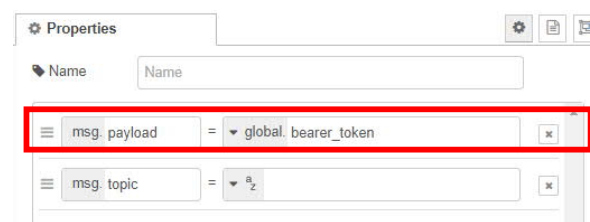


Fig. 34 Asignar la variable global

- Configurar en la sección de msg.payload como:
global.%Nombre_de_la_variable_global%.
- De esta manera, el payload que ingresa al flow corresponde al valor de la variable global, la cual posee el valor del Bearer token. Este Flow solo se activa cuando se hace clic en el nodo "inject"

5.4.3 Configuración del nodo function

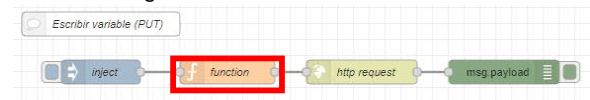


Fig. 35 Nodo function

- Haga doble clic en el nodo function

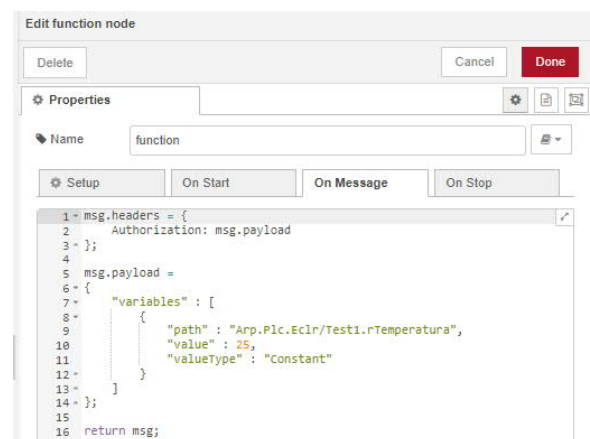


Fig. 36 Mensaje del nodo function

- "path": %VariablePath1%
- "value": %ConstantValue1% (valor de escritura)
- "valueType": "Constant"

- Es posible solicitar la escritura de más de una variable en un mismo nodo function.

```
[
  {
    "path": "%VariablePath1%",
    "value": %ConstantValue1%,
    "valueType": "Constant"
  },
  ...
  {
    "path": "%VariablePathN%",
    "value": %ConstantValueN%,
    "valueType": "Constant"
  }
]
```

Fig. 37 Agregar más variables

5.4.4 Configuración del nodo http request



Fig. 38 Nodo http request

- Haga doble clic en el nodo http request

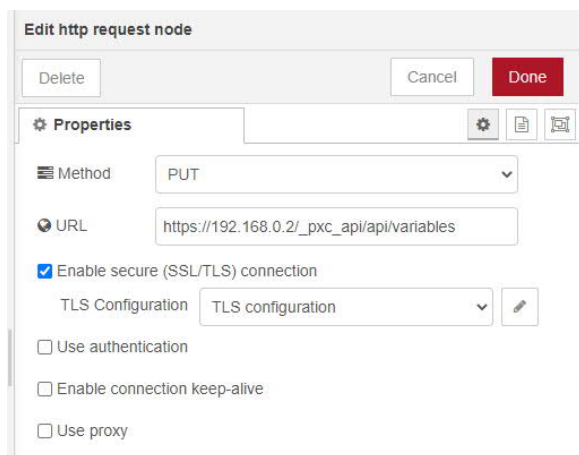


Fig. 39 Config. del nodo http request

- Método: PUT
- URL:
https://%PlcAddress%/_pxc_api/api/variables

Donde:

- %PlcAddress%: dirección IP del PLC

- Deshabilite los certificados y la opción "Verify server certificate" al igual que en el caso anterior

5.4.5 Mensaje de respuesta

- La respuesta es el código de estado HTTP 200(OK) para una lectura exitosa de la variable junto con los datos JSON resultantes; de lo contrario se lee el código de estado HTTP correspondiente (por ejemplo, 400= Solicitud incorrecta, 408= Tiempo de espera de solicitud, 500= Error interno del servidor).

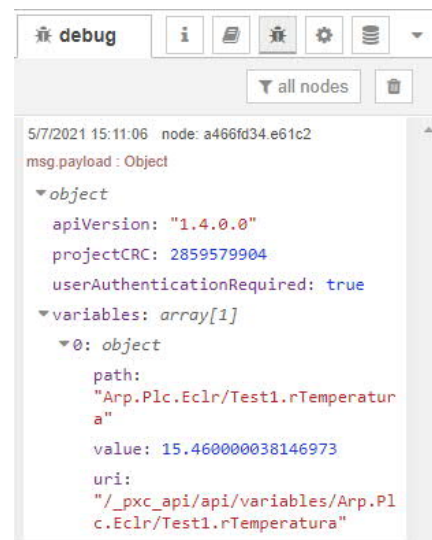


Fig. 40 Respuesta JSON