How to Use RESTful API to Connect to the ioThinx 4510 Series

Moxa Technical Support Team <u>support@moxa.com</u>

Contents

Prepare the Following Items	
Using RESTful API via Postman	
Introduction to Postman	
Install Postman	
Enable the RESTful API service on the ioThinx 4510	
Configuring Postman	
Send a GET request and a PUT request	5
RESTful API List Rules	5
Send a GET request	7
Send a PUT request	

Copyright $\ensuremath{\mathbb{C}}$ 2019 Moxa Inc.

Released on May 31, 2019

About Moxa

Moxa is a leading provider of edge connectivity, industrial networking, and network infrastructure solutions for enabling connectivity for the Industrial Internet of Things. With over 30 years of industry experience, Moxa has connected more than 50 million devices worldwide and has a distribution and service network that reaches customers in more than 70 countries. Moxa delivers lasting business value by empowering industry with reliable networks and sincere service for industrial communications infrastructures. Information about Moxa's solutions is available at <u>www.moxa.com</u>.

How to Contact Moxa

Tel: +886-2-8919-1230 Fax: +886-2-8919-1231



How to Use RESTful API to Connect to the ioThinx 4510 Series

In this tutorial you will learn how to use Restful API to send a GET request and a PUT request to the ioThinx 4510.

Prepare the Following Items

- ioThinx IIoT Starter Kit (ioThinx 4510, 45MR-1601, 45MR-2600, and 45MR-3800)
- Software: Postman, ioThinx 4510 (with firmware v1.1 installed)

Using RESTful API via Postman

Introduction to Postman

For information about Postman, please refer to the following web page:

<u>https://www.getpostman.com/</u>. Step-by-step procedures on how to use Postman are described below.







Postman(RESTful API client) IP address: 192.168.127.200 Moxa ioThinx 4510 IP address: 192.168.127.254

Install Postman

Download **Postman** from the following website and install it: <u>https://www.getpostman.com/downloads/.</u>

Enable the RESTful API service on the ioThinx 4510

1. Log in to the ioThinx 4510

Step 1: Open your web browser and type the default IP address of the device: 192.168.127.254.

Step 2: On the login page, type the default username/password (admin/moxa) to log in to the Web Console.

How to Use RESTful API to Connect to the ioThinx 4510 Series

2. Click **Security** in the left menu and then select **RESTful API via HTTP** Service Settings.

MOXA	ioThin	ix 4510				Save 8	& Restart
Dashboard		Service Settings	User Settings	Account Settings	Access Control	Certific	ate Settings
System Security	Servi	ce Settings					
letwork		No.		Service		TCP/UDP	Port
fodule		1	We	b Service via HTTP		TCP	80
erial Port		2	Web	Service via HTTPS		TCP	443
I/O Internal Register MUST import the self-signed certificate before enabling the we block the connection					https, or the browsers may		
rotocol -		3	RE	STful API via HTTP		TCP	80
Modbus		4	RES	STful API via HTTPS		TCP	443
SNMP		5		SNMP Agent		UDP	161
		6	М	odbus/TCP Slave		TCP	502
		7	Мо	odbus/RTU Master		-	-
		8		MQTT Client		TCP	-
		9	IOxpi	ress/MCC Tool/MXIO		TCP/UDP	10124/4800

3. Click **Save & Restart** in the top right corner of the page and then click **Save and Restart** in the center of the page.

Dashbaard System Security Network	MOXA [®] ioThinx 4510	Save & Restart Logout
Security	Dashboard	
	System	
Network	Security	
	Network	
Module And	Module	
Serial Port	Serial Port	
I/O	1/0	
Internal Register	Internal Register	
Protocol - Configuration has been modified.	Protocol -	Configuration has been modified
Modbus Do you want to save current setting to device?	Modbus	
SNMP Save and Restart Cancel	SNMP	Save and Restart Cancel
MQTT	MQTT	

How to Use RESTful API to Connect to the ioThinx 4510 Series

Configuring Postman

- 1. Open Postman and create a new **Request** (named **GET example**).
- 2. Create a **Collection**.
- 3. Save the new **Request** in the created **Collection**.
- 4. Create another new **Request** (named **PUT example**) and save it in the created **Collection**.

New Mimport Runner	👪 My Workspace 👻 🎄 Invite	🧿 😢 🖨 🛦 🖤 Sign In
Q Filter History Collections Treat Treat Treat Treat Treat Treat Treat Treat Treat Treat Treat Treat Treat Treat Treat Treat Treat Treat Treat Treat Treat Tre	IDDLETED] 4IT GET @ SAVE REQUEST 192.168.127.254/ Request name PUT * GET example Request description (Optional) Params Authory KEY Key Response Adding a description makes your docs better	X No Environment Send Save Cookies Code Comments(0) Description Useription
Collections liet you group related requests, making them easier to access and run.	Descriptions support Markdown Select a collection or folder to save to: Q Search for a collection or folder All Collections Create Collection Cancel	sponse.

 Select the GET method on the GET example tab, and select the PUT method on the PUT example tab.

GET GET example X PUT PUT e	example + •••	No Environment	* O			
GET example			Examples (0)			
GET v 192.168.127.254/api/io/di/GE	eT_Demo@Button/diStatus	Send	- Save -			
Params Authorization Headers (2) E	Body Pre-request Script Tests	Cookies	Code Comments			
KEY	VALUE	DESCRIPTION	••• Bulk Ed			
Key	Value	Description	Description			
GET GET example PUT PUT		No Environment	▼ ⊙			
GET GET example PUT PUT			• 💿 Examples (0)			
	example X + ••••		-			
GET GET example PUT PUT + PUT example PUT ▼ 192.168.127.254/api/io/do/P	example X + ••••	No Environment	Examples (0)			
GET GET example PUT PUT + PUT example PUT ▼ 192.168.127.254/api/io/do/P	example X + ••• PUT_Demo@Light/doStatus	No Environment	Examples (0)			

How to Use RESTful API to Connect to the ioThinx 4510 Series

- 6. Create the following content in the **Headers** of the **GET example** tab and **PUT example** tab.
 - Content-Type: application/json
 - Accept: vdn.dac.v2

Params Authorization Headers (2) Body Pr			Body Pre	Pre-request Script Tests		Cookies Code Comments (0)	
	KEY			VALUE	DESCRIPTION	••• Bulk Edit Presets 🔻	
\checkmark	Content-Type			application/json			
\checkmark	Accept			vdn.dac.v2			

Send a GET request and a PUT request

RESTful API List Rules

You can check the RESTful API list in the Appendix of the ioThinx 4510 User's Manual: RESTful API Rules.

RESTful API	Description	Access	Format
/api/io/do/{ioName}/doStatus	DO - status (0: OFF, 1: ON)	R/W	0 or 1
/api/io/do/{ioName}/doPulseCount	DO - Pulse mode - count	R/W	0 to 65535
/api/io/do/{ioName}/doPulseOnWidth	DO - Pulse mode - ON width (unit: 500us)	R/W	1 to 65535
/api/io/do/{ioName}/doPulseOffWidth	DO - Pulse mode - OFF width (unit: 500us)	R/W	1 to 65535
/api/io/do/{ioName}/doPulseStatus	DO - Pulse mode - status (0: Stop 1: Start)	R/W	0 or 1

The {ioName} is composed of three elements: Module_Name, @, and I/O_Channel_Name. For example, the {ioName} associated with module name "45MR-2600-0" and DO channel name "DO-00" is 45MR-2600-0@DO-00.

Note: The module and I/O channel name can be changed in Module Settings and I/O Settings.

How to Use RESTful API to Connect to the ioThinx 4510 Series

1. In this demonstration, we change the name of module **45MR-1601** to **GET_Demo** and the name of **45MR-2600** to **PUT_Demo**.

MOXA	Save & Restart Logout							
Module	Module Se							
	Match	Click finish button to con	nplete module settings.					
	🛷 Save	e Settings × Cancel	s (
	Slot	Detected Module	Configured Module					
	1	45MR-1601 →	45MR-1601 GET_Demo 💼 =					
	2	45MR-2600 →	45MR-2600 PUT_Demo					
	3	45MR-3810 →	45MR-3810 45MR-3810-0					

2. Change the **DI-00** channel name to **Button**.

	oThinx 451	10				Logout
Dashboard	I/O Settings					
System	i/O Settings					
Security	GET_Demo	~				
Network						
Module	DI-00	DI	٠	Button	•	
Serial Port						
1/0	DI-01	DI		DI-01	•	
Internal Register						
Protocol +	DI-02	DI	•	DI-02	•	

3. Change the **DO-00** channel name to **Light**.

	oThinx 4510	Save & Restart	Logout
Dashboard	I/O Settings		
System	ir o occurrigo		
Security	PUT_Demo 👻		
Network			
Module	DO-00 DO • Light	•	
Serial Port			
I/O	D0-01 D0 • D0-01	•	
Internal Register			
Protocol +	D0-02 D0 • D0-02	*	

Copyright © 2019 Moxa Inc.

How to Use RESTful API to Connect to the ioThinx 4510 Series

Send a GET request

1. Enter the request URL based on the IP address, file type, and API map. For this example, the request URL should be written as:

192.168.127.254/api/io/di/GET_Demo@Button/diStatus

2. Click the **Send** button. You should see the status code, response time, and results.

GET GET example X PUT PUT example	• + •••	No Environment 🔻 📀 暮						
GET example 2 Examples (0) •								
Params Authorization Headers (2) Body Pre	-request Script Tests	Cookies Code Comments (0)						
KEY	VALUE	DESCRIPTION *** Bulk Edit Presets 🔻						
Content-Type	application/json							
Accept	vdn.dac.v2							
Key	Value	Description						
Body Cookies Headers (3) Test Results	ody Cookies Headers (3) Test Results Status: 200 OK Time: 70 ms Size: 113 B Save Download							
Pretty Raw Preview JSON - 5								
1 * { 2 3 } 1 * value": 0								

3. Press the DI0 button on the starter kit to trigger the DI-00 channel, and then click **Send** again. In response, the DI value will change.

Body Cookies Headers (3) Test Results	Status: 200 OK Time: 56 ms Size: 113 B Save Download
Pretty Raw Preview JSON 🔻 🚍	■ Q
1 ~ { 2 "value": 1	
3 }	

How to Use RESTful API to Connect to the ioThinx 4510 Series

Send a PUT request

1. Enter the following request URL:

192.168.127.254/api/io/do/PUT_Demo@Light/doStatus

- 2. Locate the **Body** tag, select raw, and choose the **JSON** format.
- 3. Enter the following message:

1 -	{		
2	-	"value":	1
3	}		

4. Click **Send**. The DO-00 channel should respond by turning the light on.

PUT example 4 Examples (0) PUT 192.168.127.254/api/io/do/PUT_Demo@Light/doStatus Send Send Save Params Authorization Headers (2) Body Pre-request Script Tests Cookies Code Comments (0) • none • form-data • x-www-form-urlencoded • raw • binary JSON (application/json) • Beautify 3 1 × (f) • form-data • x-www-form-urlencoded • raw • binary JSON (application/json) • Beautify	GET GET e	example		PUT PUT example		• +	••••			No En	vironment	Ŧ	٩	\$
Params Authorization Headers (2) Body Pre-request Script Tests Cookies Code Comments (1) none form-data x-www-form-urlencoded raw binary JSON (application/json) Beautify <!--</td--><td>▶ PUT exa</td><td>ample</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>4</td><td>Exa</td><td>amples (0)</td><td>) -</td>	▶ PUT exa	ample	1								4	Exa	amples (0)) -
none form-data x-www-form-urlencoded raw binary JSON (application/json) Reautify	PUT	*	192.168.127.254/a	api/io/do/PUT_Demo	o@Light/do	Status					Send	•	Save	*
	Params	Authoriz	ation Headers	; (2) Body •	Pre-req	uest Script	Tests				Cookies	Code		
3 a - [r]		• form	-data 🔍 x-www	-form-urlencoded	raw	binary	JSON (ap	plication/json)	•				Beau	tify
2 "value": 1 3)	2	"value":	1											