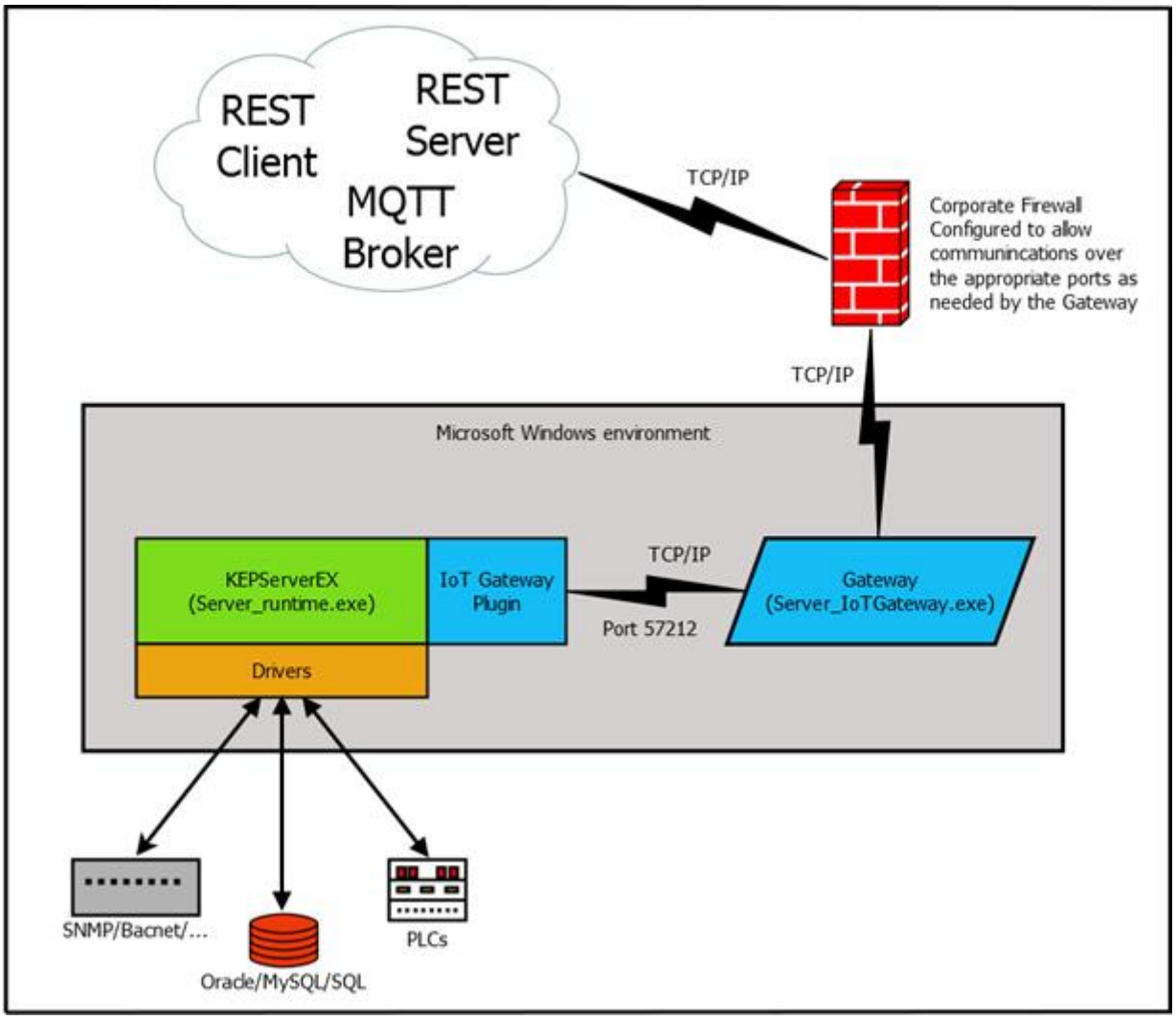
kepware®

KEPServerEX v6 - IoT Gateway 操作說明

IoT Gateway Plug-In 是 KEPServerEX 的可選功能，IoT Gateway 可將系統及設備上的 tag 資料，透過符合標準的 IP 協定，發送至第三方 endpoints，當 tag value 改變或達到設定的 Sacn Rate，IoT Gateway 便將 tag ID、value、quality、timestamp，以標準 JSON 格式發送至對應的第三方 endpoints。IoT Gateway Plug-In 提供以下功能：

- 可發送資料來源伺服器中任何資料，如驅動程式、套件、系統數值變化，包含 tag ID、value、quality、timestamp。
- 標準且易讀的 JSON 資料格式。
- 可透過 MQTT，REST client 發送，以及透過 REST server agents 作雙向通訊。
- REST 及 MQTT Client，可設定接收及發送速率，範圍為 10 毫秒/次 ~ 27.77 小時/次。
- 支援所有 Agent 認證及 TLS / SSL 加密。
- 支援 KEPServerEX 使用者及 Security Policies Plug-In 使用者權限管理。
- 可針對不同的第三方 endpoints，方便的設定資料標頭及乘載資訊。

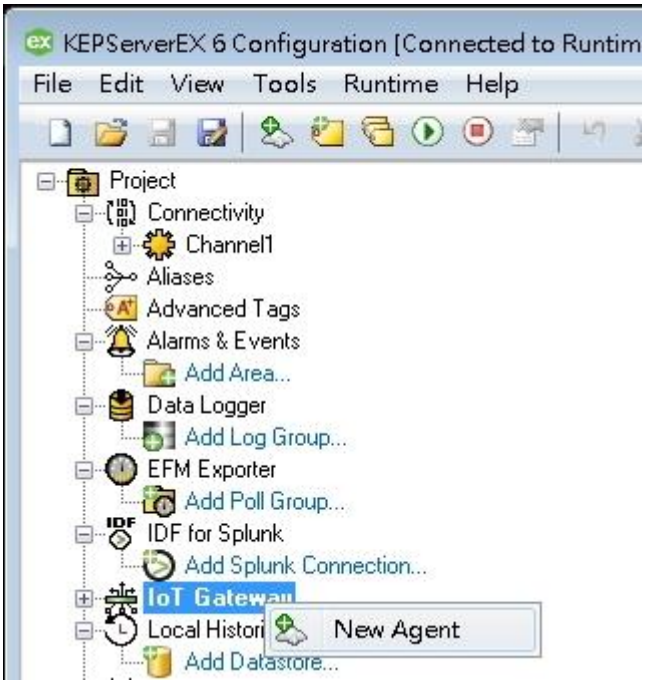
IoT Gateway Plug-In 系統架構圖：



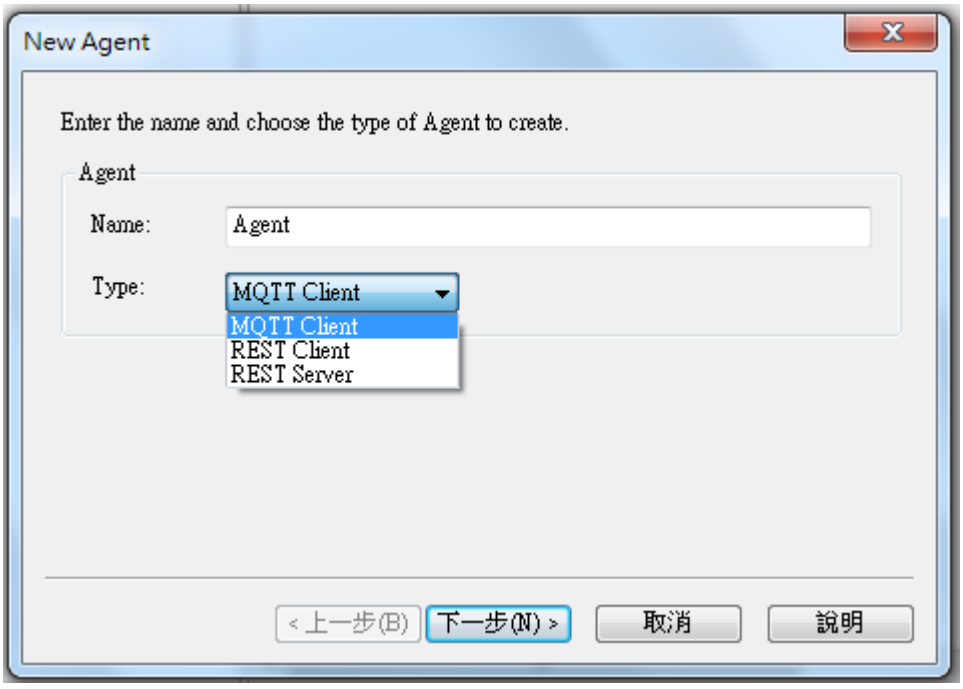
快速入門操作

新增 IoT Gateway 連接

1. 在左方樹狀選單的 IoT Gateway 項目，點擊滑鼠右鍵，點擊 New Agent。



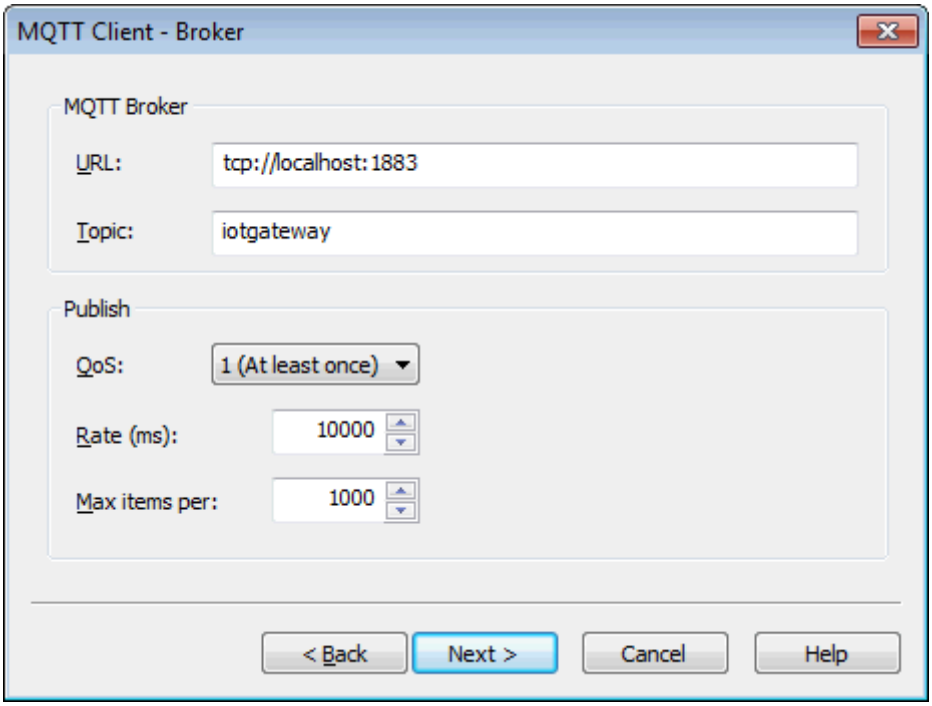
2. 在新增 Agent 視窗中，輸入一個 Agent 的 Name，可以自行定義，並選擇類型： **MQTT Client**、**REST Client** 或 **REST Server**。



設定 MQTT 連接

設定 Agent 類型為 MQTT Client，請按照下列步驟來建立一個新的 MQTT Agent 連接：

1. 在 MQTT Client 視窗中設定 broker Server 對應。



URL: 這是 Agent 連接 endpoints 的 IP 地址或 URL 和 port，預設：tcp://localhost:1883

Tip: 如果 endpoints 使用 SSL 連接，調整 URL 中使用“SSL：//”，而不是“TCP：//”。

Topic:這是用來過濾或安排發送數據的 broker 名稱。

QoS: 這是 MQTT 設置發佈服務的數值。選擇包括：0（最多一次），1（至少一次），2（恰好一次）。

Rate（毫秒）：設置 Agent 發送資料至 endpoints 的速率。範圍是 10 至 99999990 毫秒。

Max. items per：在 Gateway 發送一次的傳輸中，item 的數量。

2. 按 **Next 下一步**。

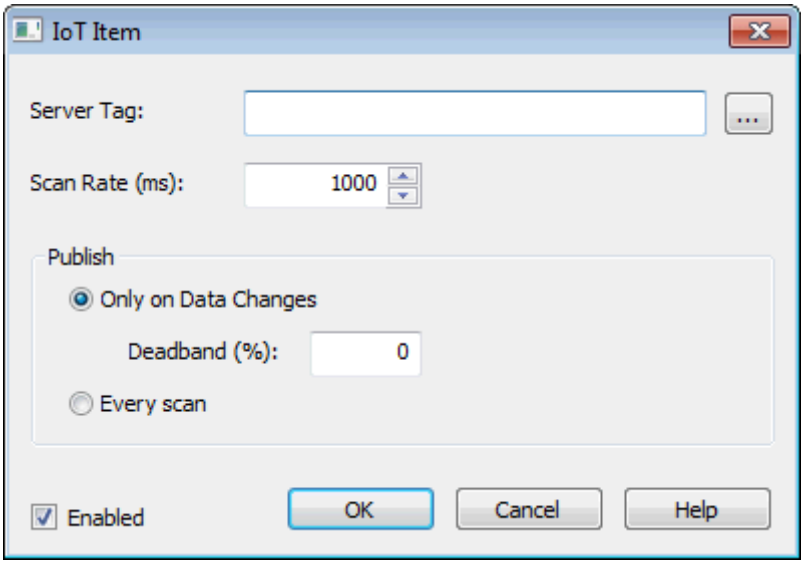
3. 若您欲連接的 Broker 有 ID 及帳號密碼驗證，便可在此設定。

4. 設定新增 Tag 。

在 Agent 新增 Tags

當你有一個 Agent 設定，你可以新增 Tag。按照下面的步驟新增一個或多個 Tag。

新增單個 Tag



Server Tag：輸入 tag 的完整 channel.device.name 或瀏覽找到一個 Tag。

Scan Rate：檢查該 tag 數值更新的速率，單位為毫秒。

Only on Data Changes：當該 tag 數值變化時，才發送資料。

Deadband:：判斷數值變化的百分比。變化是基於所有範圍 Tag 的數據類型。若輸入 0，表示有變化的數值皆發送出去

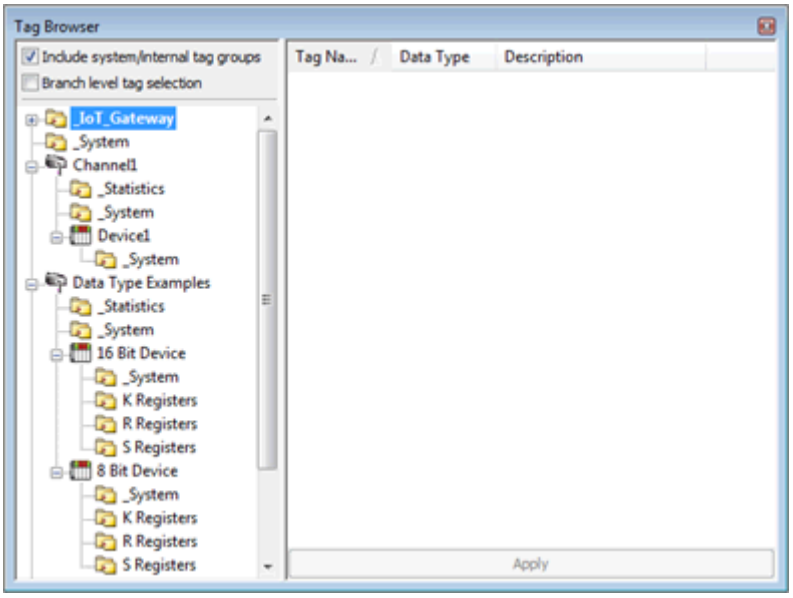
Every Scan：不論數值有無變化，皆發送資料。

Enabled：允許或禁止發送此 tag 資料。未啟用仍計入 tag 授權數。

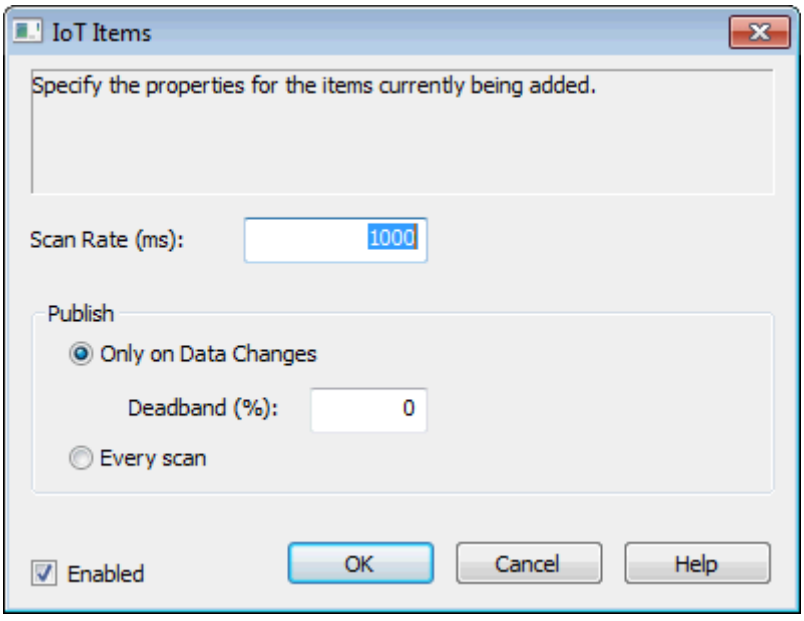
按 **OK**。

新增多個 Tags

1. 在 Tag Browser 新增點選 Tag。



4. 可以點選多個 Tag 後，按 **Apply**。



3. 按 **OK** 。

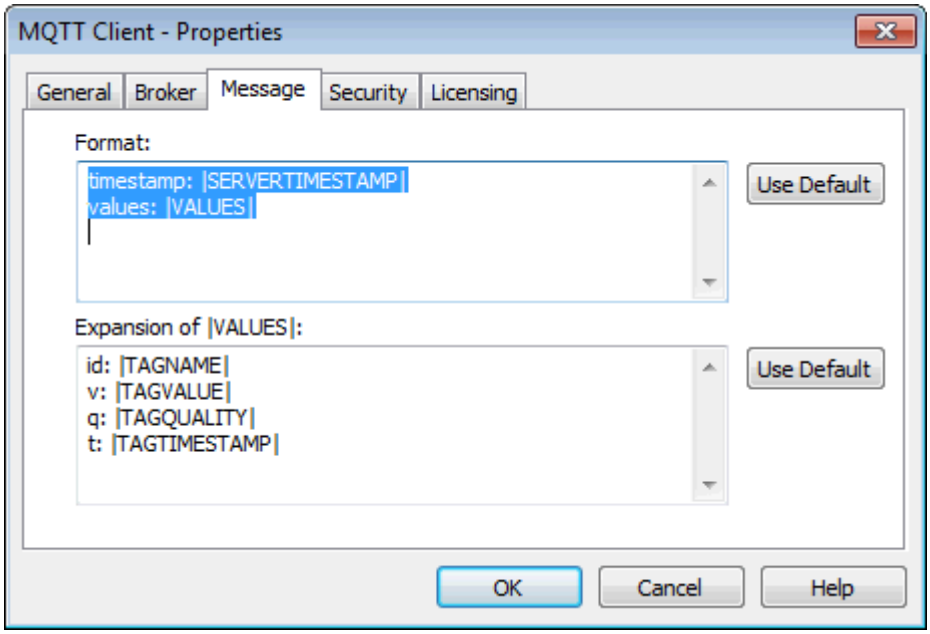
MQTT Client Message 訊息管理

要改變 JSON 數據加載順序或刪除數據項，請按照下列步驟操作：

1.雙擊或 Agent 名稱右鍵點擊並選擇 **Properties**，打開 **Properties** 視窗。

2.在屬性視窗中，選擇 **Message** 選項。

3.進行必要的更改（根據以下準則），然後點擊確定。



Format：包含任何在 JSON 乘載之前或之後的資料。每發送一次，便被送出一次。格式為名稱加上冒號加上數值。也可包含靜態文字的鍵值。有效數值變化是：

| SERVERTIMESTAMP | Gateway 發送資料到 endpoints 的日期和時間，以 UNIX 或 POSIX 的時間格式發送。

| SERVERDATE | 一樣是 Gateway 發送資料的日期和時間戳記，但以人類易讀的格式發送。

| VALUES | 依下面區塊來結合乘載。

點擊 Use Default 按鈕將資料及格式恢復至預設值。

擴大 | VALUES |：包含了 JSON 乘載並發送資料到 endpoint 的格式。這些的 Values 可以被重新排序或移除。發送一次資料至 endpoints 中，可包含多個 instances。例如，在兩次發送資料之間，1 個 tag 有 4 個資料改變，那麼在該陣列中便包含 4 個完整的 JSON 字串。在 Format Box 中，每個| VALUES |前可輸入您想要的文字或敘述。冒號則是區分名字與數值。有效數值變化是：
| TAGNAME | 該 tag 的名稱

| TAGVALUE | 該數值變化的值

| TAGQUALITY | 指該 tag 的 quality，為 good 或 bad

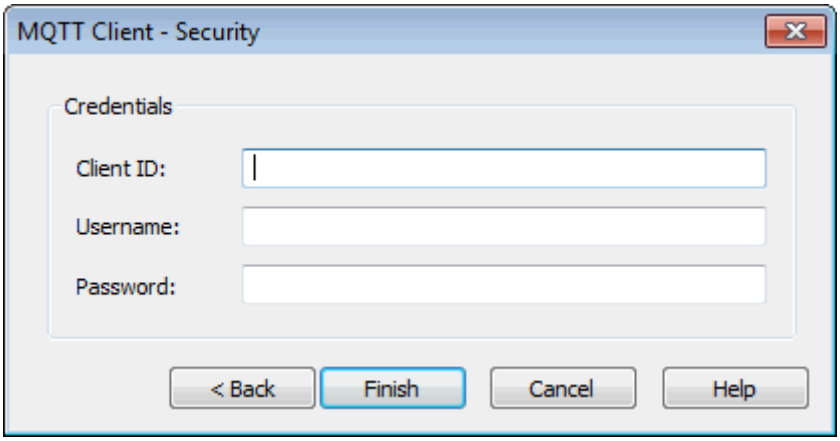
| TAGTIMESTAMP | 指接收到 tag value 的時間

點擊 Use Default 按鈕將資料及格式恢復至預設值。

注：每個表達式需要包含一個名稱與對應值。

MQTT Client Security 安全性設定

按照下面的步驟來設定 MQTT Agent 安全性授權和驗證：
1.在安全性視窗中，輸入下面的憑證：



Client ID: 這是針對 Agent 連接的 MQTT broker，大多數 brokers 不需要 Client ID 來連接。

Username: 這是有授權 broker 的使用者

Password: 這是有授權 broker 的使用者密碼

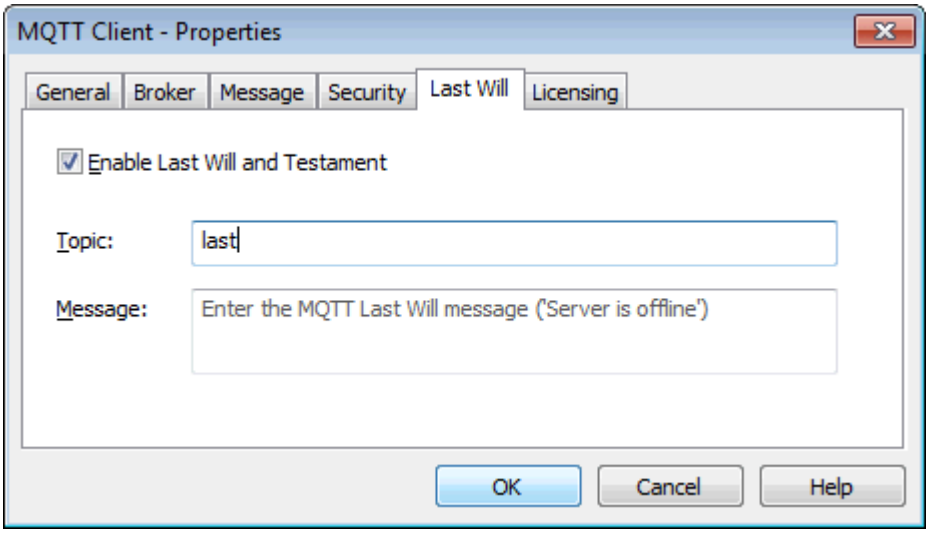
Note: 如果不使用 SSL 加密連接，使用者名和密碼以純文字發送到 broker。這是協定的限制

2. 按 **Finish** 完成。
3. 可以繼續新增 Tags。
4. MQTT 連接現在發佈給 Agent。驗證 Agent 接收。如果沒有，檢查事件日誌中的錯誤。

MQTT Last Will and Testament

MQTT Last Will
該“Last Will”（LWT）是 MQTT 通知訂閱的客戶，當客戶端沒有“斷開”通知意外斷開連接的約定。要啟用和設定 MQTTLast Will，請按照下列步驟操作：
對 Agent 名稱

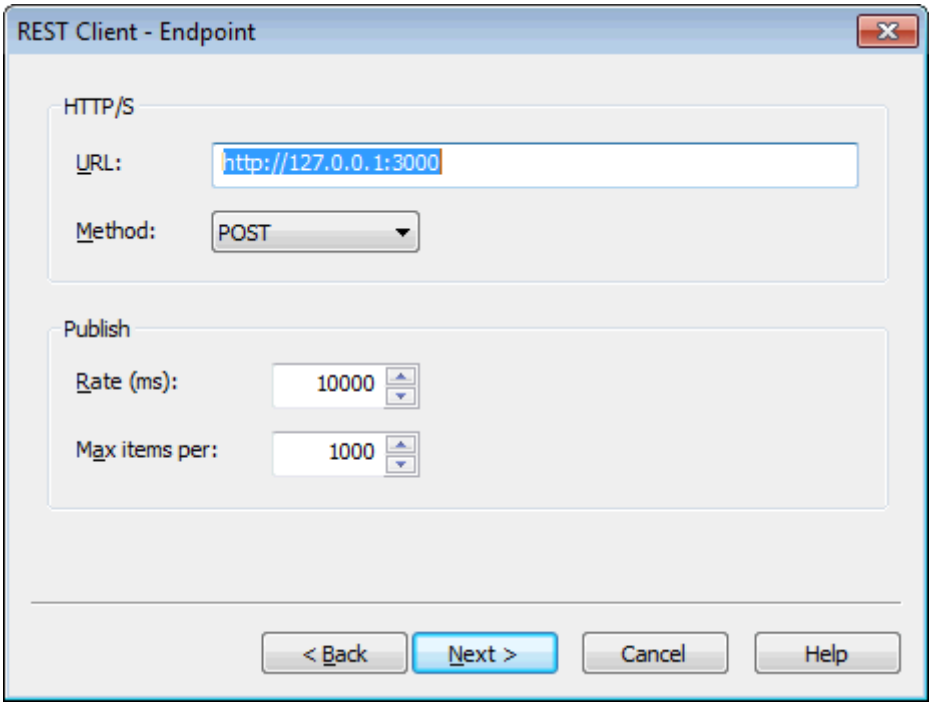
1.雙擊或 Agent 點擊鼠標右鍵，選擇屬性，打開屬性視窗。
2.在屬性視窗中，選擇最後將選項卡。
3.點擊啟用 Last Will 複選框打開了“最後 Last Will”的消息。
4.Topic 或路徑作為 LWT。
5.輸入訂閱的客戶端將收到的 LWT 消息的文本。這是典型的非正常斷開連接的說明，如“當機”或“意外退出”，以幫助客戶。
6.點擊 OK。



設定 REST Client 連接

設定 Agent 類型選擇為 REST Client，請按照下列步驟來建立一個新的 REST Client Agent 連接。

1. 設定 REST Client Endpoint 參數。



URL：輸入欲連結 endpoint 的 IP address 或 URL 及 port，如果 endpoint 使用 SSL 連接，則 URL 必需使用“https://”

Method：發送數據到 endpoints 的方式，可以是 POST 或 PUT 指令。

Publish Rate：發送資料到 endpoints 的速率。

Max. items per：在 Gateway 發送一次的傳輸中，item 的數量。

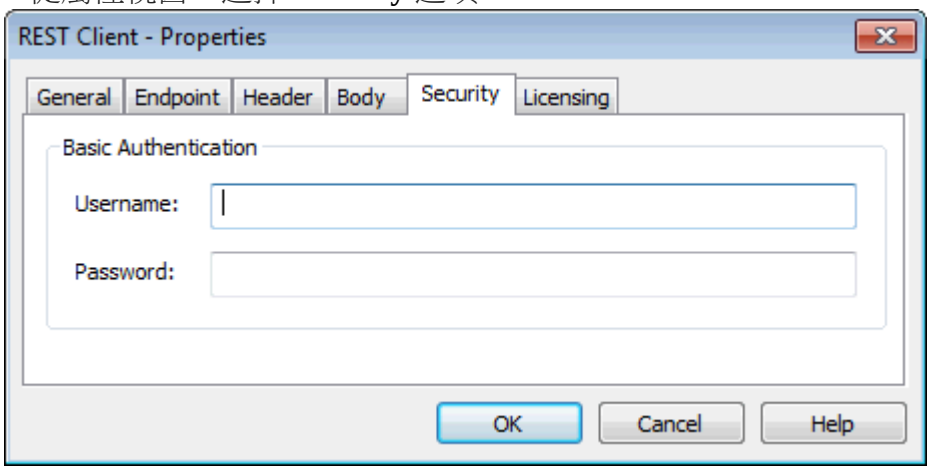
按 Next 下一步。

REST Client Security 安全性設定

如果 REST Client 需要基本身份驗證，請按照下列步驟操作：對 Agent 名稱

1.雙擊或 Agent 單擊鼠標右鍵，選擇屬性，打開屬性視窗。

2.從屬性視窗，選擇 **Security** 選項。



Username：輸入基本的 HTTP 驗證有效的使用者名稱。

Password：輸入與指定的使用者名稱關聯的密碼。

3.進行對應，然後單擊 **OK** 確定。

Adding Tags to an Agent

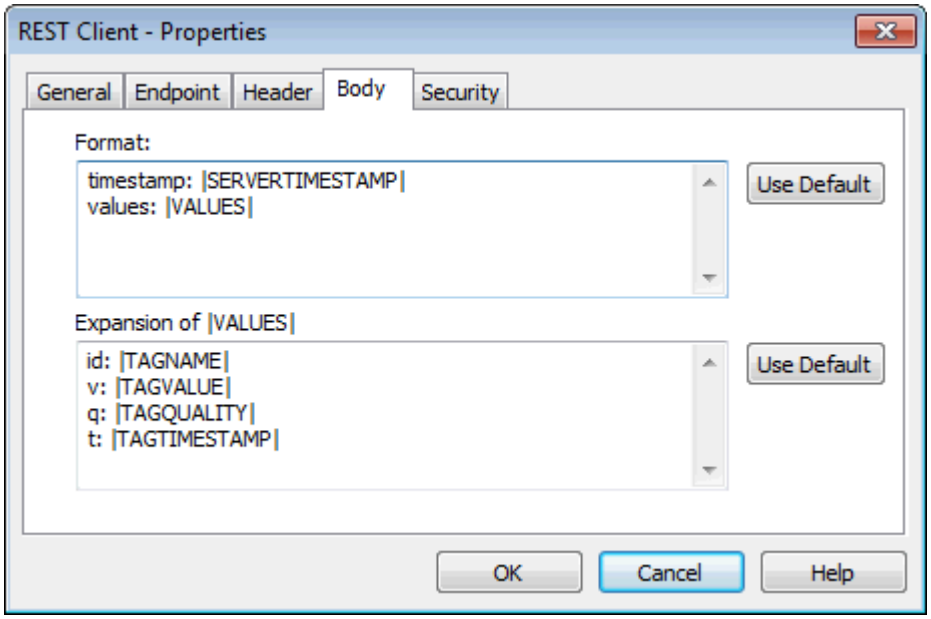
當你有一個 Agent 設定，你可以新增 Tag。按照下面的步驟新增一個或多個 Tag。

可以加參考 MQTT Client 新增 Tag 設定

REST Client Body

要改變 JSON 數據加載順序或刪除數據項，請按照下列步驟操作：
對 Agent 名稱

1.雙擊或 Agent 單擊鼠標右鍵，選擇屬性，打開屬性視窗。
2.在屬性視窗中，選擇 Body 選項卡。
3.進行必要的更改（根據以下準則），然後單擊確定。



Format：包含任何在 JSON 乘載之前或之後的資料。每發送一次，便被送出一次。格式為名稱加上冒號加上數值。也可包含靜態文字的鍵值。有效數值變化是：

| SERVERTIMESTAMP | Gateway 發送資料到 endpoints 的日期和時間，以 UNIX 或 POSIX 的時間格式發送。

| SERVERDATE | 一樣是 Gateway 發送資料的日期和時間戳記，但以人類易讀的格式發送。

| VALUES | 依下面區塊來結合乘載。

點擊 Use Default 按鈕將資料及格式恢復至預設值。

擴大 | VALUES |：包含了 JSON 乘載並發送資料到 endpoint 的格式。這些的 Values 可以被重新排序或移除。發送一次資料至 endpoints 中，可包含多個 instances。例如，在兩次發送資料之間，1 個 tag 有 4 個資料改變，那麼在該陣列中便包含 4 個完整的 JSON 字串。在 Format Box 中，每個| VALUES |前可輸入您想要的文字或敘述。冒號則是區分名字與數值。有效數值變化是：
| TAGNAME | 該 tag 的名稱

| TAGVALUE | 該數值變化的值

| TAGQUALITY | 指該 tag 的 quality，為 good 或 bad

| TAGTIMESTAMP | 指接收到 tag value 的時間

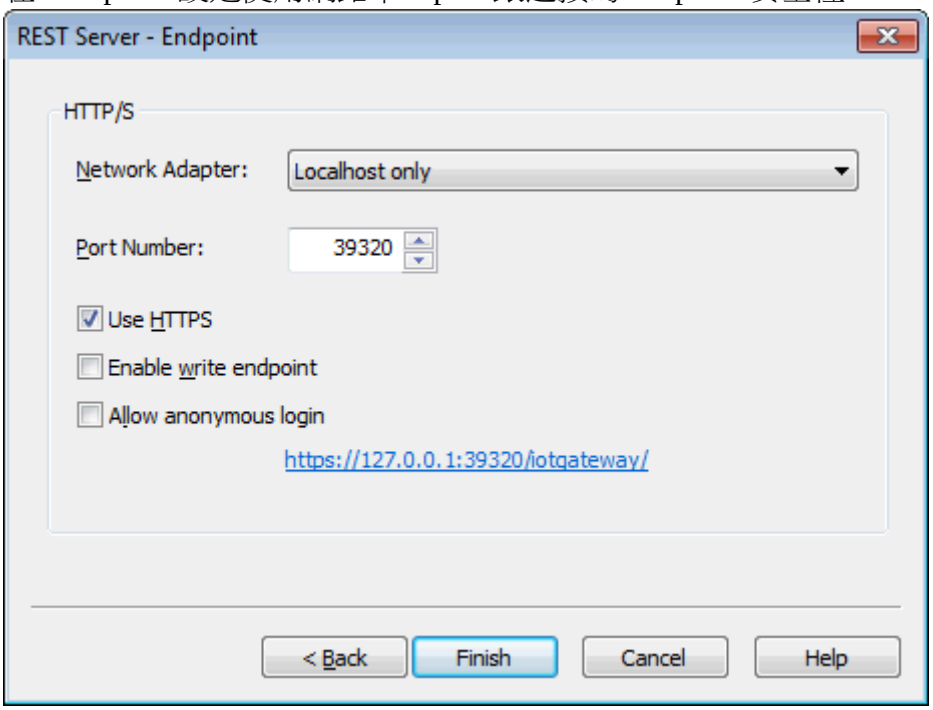
點擊 Use Default 按鈕將資料及格式恢復至預設值。

注：每個表達式需要包含一個名稱與對應值。

設定 REST Server 連接

一旦設定 Agent 類型選擇為 REST Server，請按照下列步驟來建立一個新的 REST Server Agent 連接。

1. 在 Endpoint 設定使用網路卡、port 跟連接的 endpoint 安全性。



Network Adapter：選擇欲使用的網路卡，預設為 localhost 或 127.0.0.1。

- **Port Number**：這是在 REST 的 server 綁定的 port。如果在同一網路卡上設定多個 REST server Agents，他們每個人都需要一個不同的 port 號。
- **Use HTTPS**：使用 HTTPS 加密 Client 與 REST server 間傳輸的資料。可自動建立憑證，也可匯入您自訂的憑證。預設為啟用。
- **Enable write endpoint**：無論 endpoints 使用者權限為何，皆允許寫入 tag 數值。啟用時，那些被指定為讀/寫的 Tag，可以根據使用者的權限寫入。如果允許匿名登錄，則所有使用者皆可讀/寫 Tag。如果不允許匿名登錄，則依照使用者讀寫權限（基於使用者管理和或安全策略組件）。預設為不勾選。
- **Allow anonymous login**：允許匿名登錄，預設所有 Client 連接必須有相對應的有效帳戶的使用者管理器或安全策略組件的憑證。如果啟用，則允許未經驗證的使用者登入。使用者管理和安全策略組件都是由 server 管理員設置。預設為不勾選。
- 在視窗底部的 URL 訪問 REST server 是動態的，依您的設定改變。

  注：本次專案的 URL 連接顯示的地址，一旦更改應用。點擊該 URL 之前被施加的變化可能導致無法加載頁。

2. 設定完畢後，單擊 Finish（完成）。

3. 下一個步驟可新增 tag。

4. 透過使用標準 HTTP 的瀏覽器，來與 REST server 連接，請求瀏覽和讀取，REST 命令格式為：

   在瀏覽器輸入 URL(瀏覽器不建議使用 IE，請用 Chrome)

   http://localhost:39320/iotgateway/browse

   http://localhost:39320/iotgateway/read?Tags=<TagName>

Adding Tags to an Agent

當你有一個 Agent 設定，你可以新增 Tag。按照下面的步驟新增一個或多個 Tag。

可以加參考 MQTT Client 新增 Tag 設定

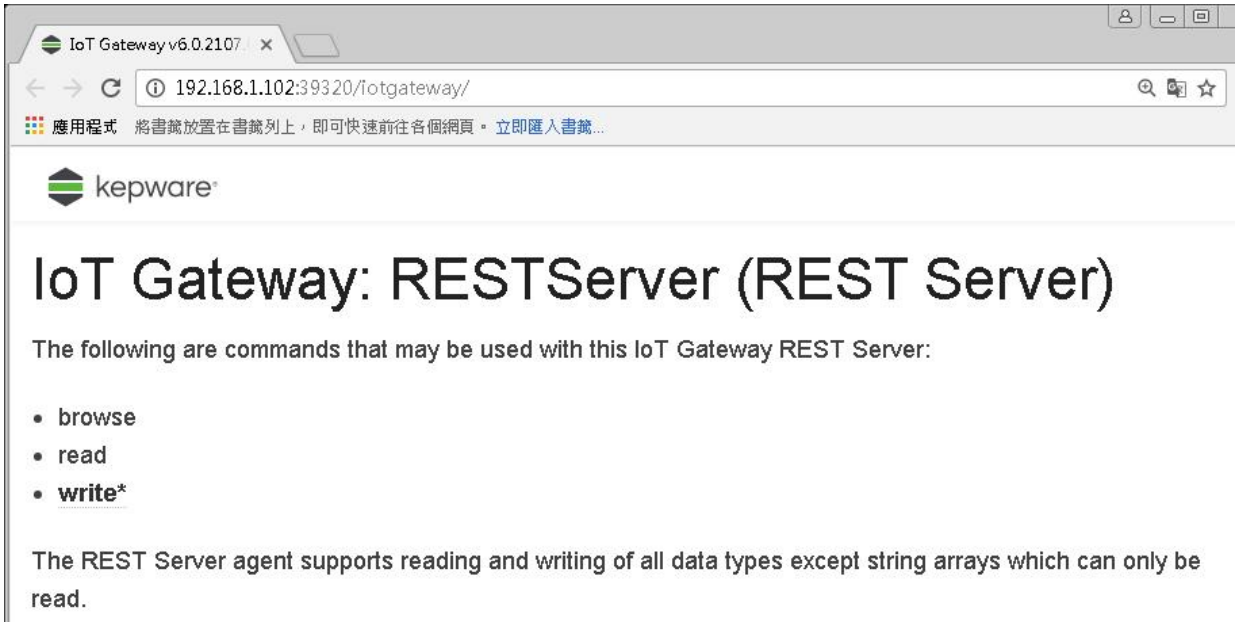
Working with a REST Server

在一個REST server Agent創立Client可以連接到endpoint瀏覽,讀取和寫入該Agent設定的tags。

REST Server Agent 核對所有連接的使用者，除非允許匿名登入，在 Agent 選項啟用憑證驗證。

有關設定使用者管理的信息，請參閱 KEPServerEX Help。使用命令來快速查詢，可透過 Web 瀏覽器導引至 endpoint。

如果 Agent 設定皆為預設值，該連接為 https://localhost:39320/iotgateway/



如果 Agent 無使用預設值，打開 REST Server Agent 的屬性，單擊 endpoints 選項，可查看目前的設定。

在 IoT Gateway Plug-in 支援多個 REST Server Agent，但需使用不同的 ports。

這 REST server 支援下列指令：

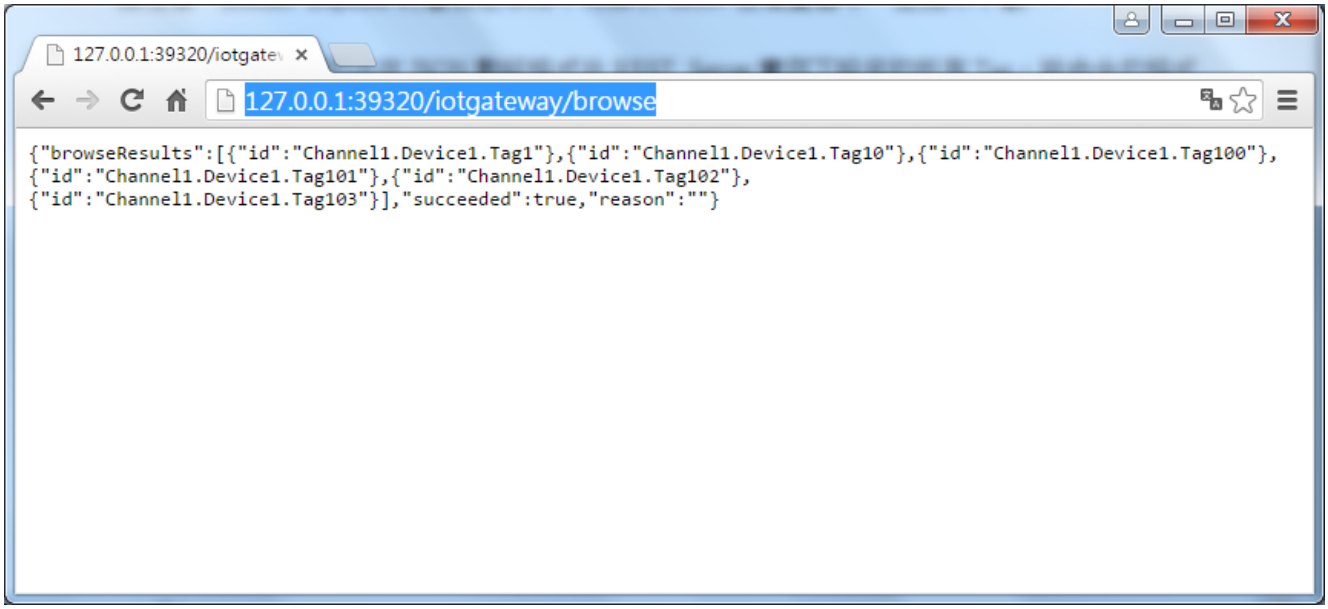
- Browse 瀏覽
- Read 讀取
- Write 寫

下面的 GET 命令可以在大多數 Web 瀏覽器進行測試。

請注意，Internet Explorer 當前版本將不在瀏覽器中解析 JSON，並提示下載。

一個瀏覽器命令將列出在 JSON 數組格式此 REST Server 實例下設定的所有 Tag。該命令的格式是：

https://localhost:39320/iotgateway/browse

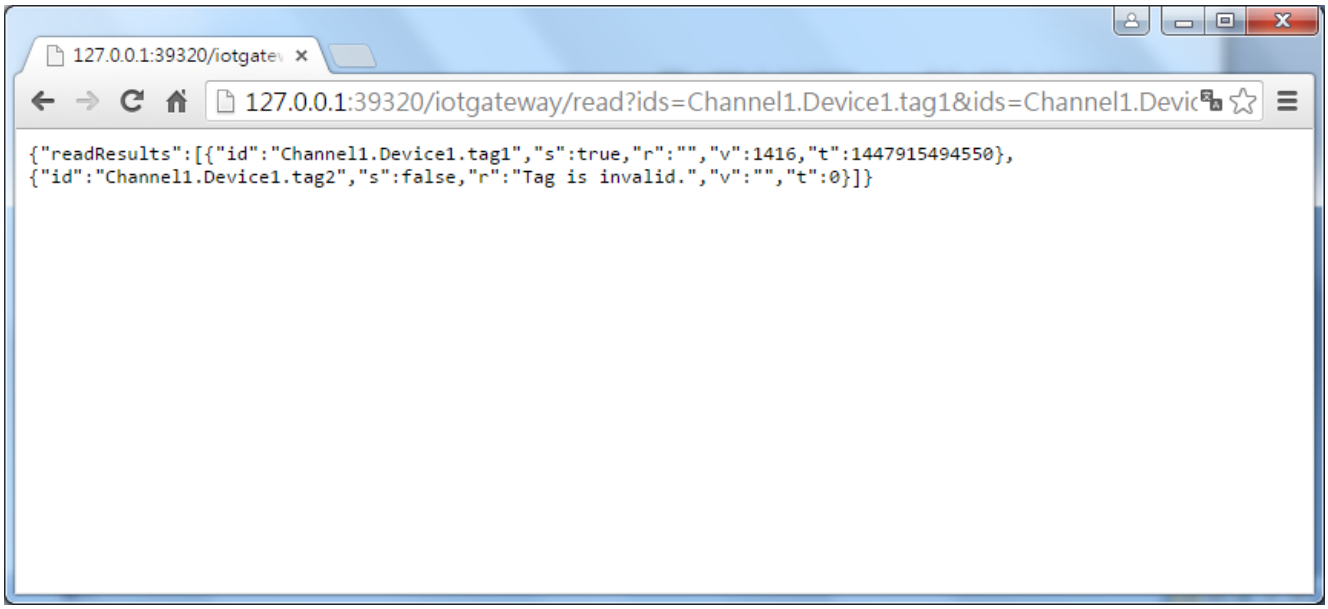


讀取命令將返回一個 JSON 數組格式要求的 Tag 或 Tags。該命令的格式是：

https://localhost:39320/iotgateway/read?ids=Channel1.Device1.Tag1

對於多個重複讀取通過與列表分隔的 Tag：

https://localhost:39320/iotgateway/read?ids=Channel1.Device1.Tag1&ids=Channel2.Device2.Tag2



以下是 POST 命令，將需要使用更專業的 Client。

在讀取 post 命令將返回一個 JSON 數組格式要求的 Tag 或 Tags。該命令的格式是：

https://localhost:39320/iotgateway/read

包含以下格式所需 Tag 的 POST 的 body：

["Channel1.Device1.Tag1”]

或多個 Tag：

["Channel1.Device1.Tag1","Channel2.Device2.Tag2","Channel3.Device3.Tag3"]

寫入命令允許一個 Client 來寫入，屬於 Client 設定下的一個或多個 Tag。為了能夠寫入 Tag，Agent 必須設定為允許寫入，並且 Tag 也必須是可寫入的。該命令的格式是：

https://localhost:39320/iotgateway/write

用含有所需 Tag 或 tags 和數值按以下格式的 POST 的 body：

[{"id": "Channel1.Device1.Tag1","v": "123"}]

對於多個 Tag：

[{"id": "Channel1.Device1.Tag1","v": "123"},{"id": "Channel2.Device2.Tag2","v": "456"},{"id": "Channel3.Device3.Tag3","v": "789"}]

在 body“ID”：是 Tag 的名稱，“V”：是寫入的值。

提示：瀏覽器導引到 https://localhost:39320/iotgateway/ 或 http://localhost:39320/iotgateway/ ，如果你禁用了 HTTPS，將為您提供這些命令的簡要描述。

注：

1.在上述範例中的 port 和 Tag 名稱需要對應到 REST Server 的設定。

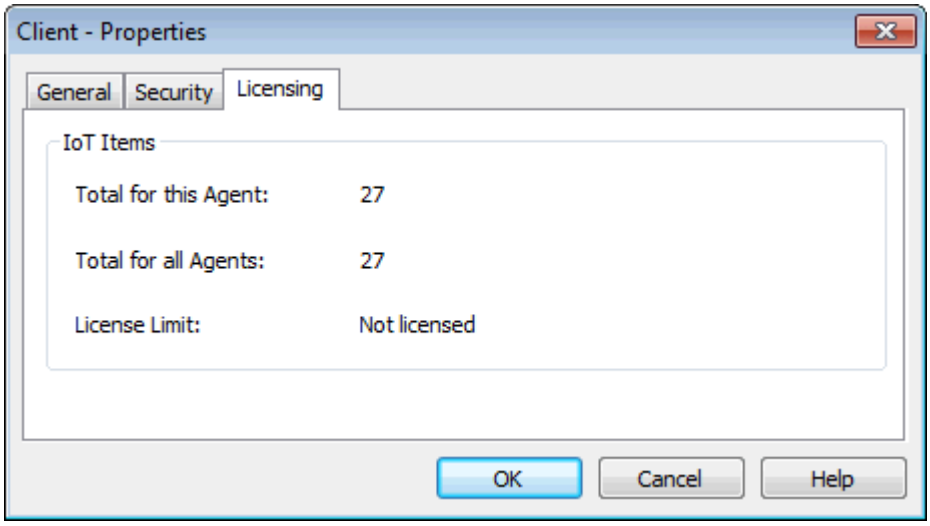
2.電腦名稱不被視為有效的 endpoints，並導致 HTTP 500 錯誤。

Licensing 授權

IoT Gateway 組件使用分層，基於計算點數的 License 模式。授權可以購買，使產品運行無限量的時間 Tag 固定的最大數量。不會阻止加入超出授權數量的 Tag，也不會發出進入 DEMO 模式的訊號，但超出授權數量的 tag，將不會更新任何 tag 的 value。

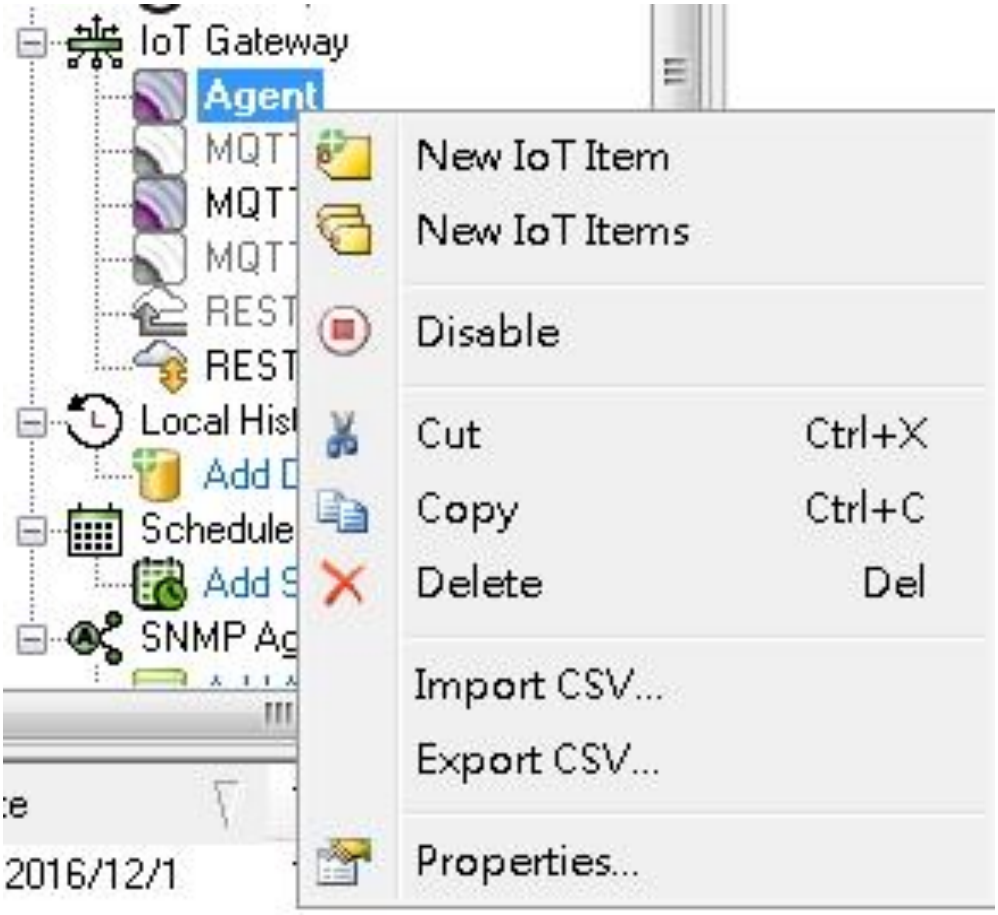
Tag 授權的數量顯示在每個 Agent 的屬性授權頁籤上。沒有為 Tag 的數量為 Agent 以及 Tag 行例在 IoT Gateway 以及授權限制的總數列表。

該授權 Tag 列出了根據原本 Agent，以及在所有 Agent 和授權級別設定的總數設定 Tag 的總數。每個 Agent 人都有在性能授權 Tag。在這個例子中，客戶端是被設定的唯一 Agent，它有設定在 Tag 視窗有 27 tag。



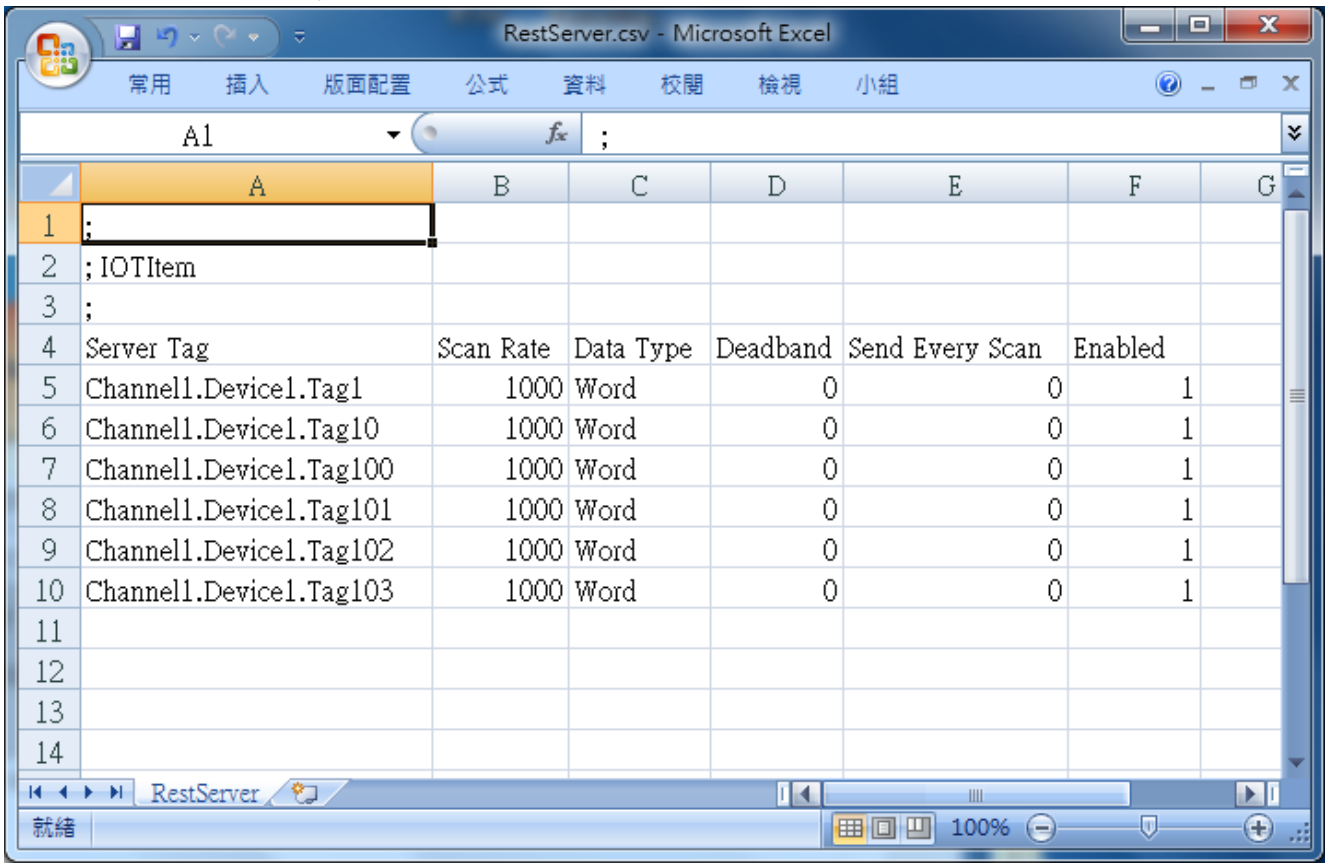
匯入跟匯出 CSV 檔案編輯

1. IoT Gateway 組件支援可以把 Agent 的 Server Tag 設定檔匯出跟匯入，透過 CSV 檔案，可以使用 Excel 或其它程式編輯修改、新增，但是沒有辦法修改 Agent 的連接設定。
2. 點選新增好的 IoT Gateway Agent 連接，按滑鼠右鍵，點選 **Export CSV**，選擇要存檔案的路徑跟檔案名稱。



匯出 CSV 檔案格式

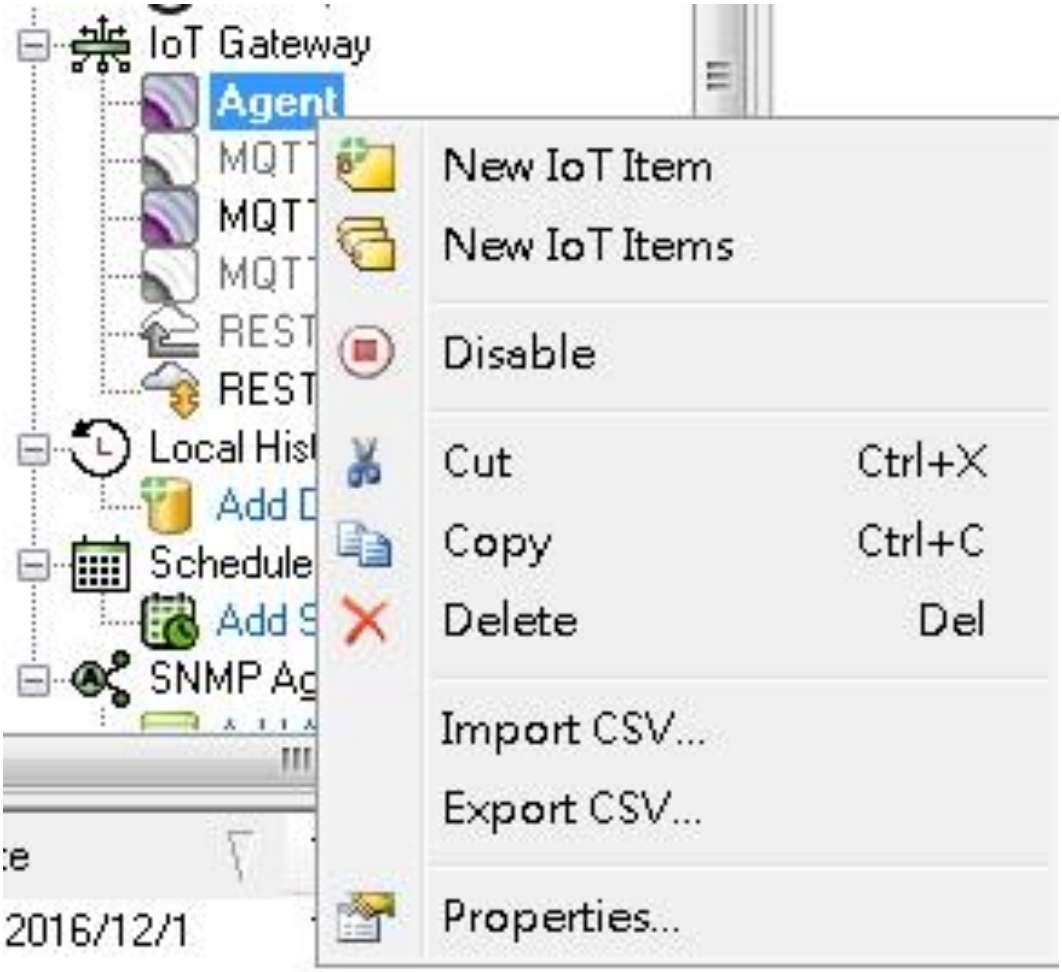
3. 匯出 CSV 可以修改 Server Tag、Scan Rate 更新速率，Data Type 資料型態，Deadband%變化百分比，Enabled 啟用狀態。



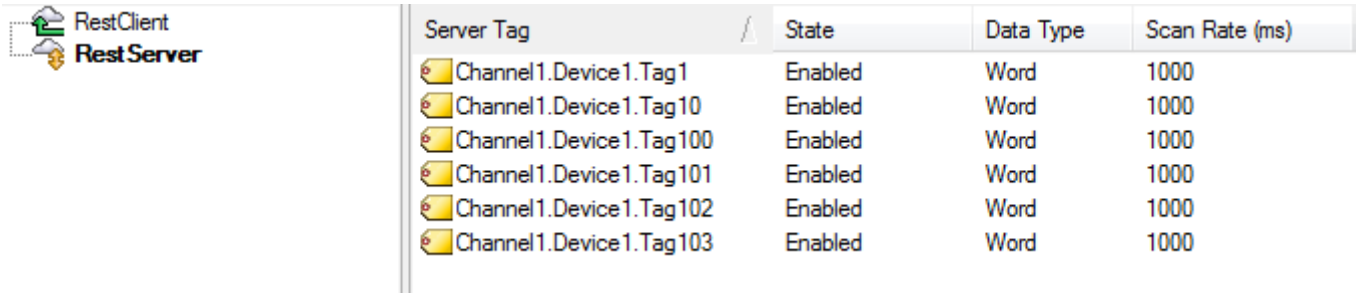
	A	B	C	D	E	F	G
1	;						
2	; IOTItem						
3	;						
4	Server Tag	Scan Rate	Data Type	Deadband	Send Every Scan	Enabled	
5	Channel1.Device1.Tag1	1000	Word	0	0	1	
6	Channel1.Device1.Tag10	1000	Word	0	0	1	
7	Channel1.Device1.Tag100	1000	Word	0	0	1	
8	Channel1.Device1.Tag101	1000	Word	0	0	1	
9	Channel1.Device1.Tag102	1000	Word	0	0	1	
10	Channel1.Device1.Tag103	1000	Word	0	0	1	
11							
12							
13							
14							

匯入 CSV 檔案

4. 點選新增好的 IoT Gateway Connection，按滑鼠右鍵，點選 **Import CSV**，選擇要匯入 CSV 檔案的路徑位置跟檔案名稱。



5. 匯入 CSV 檔案後，就能看到匯入的 Server Tag。



REST Server / Client 使用範例

REST Server 供 endpoints 瀏覽、讀取、寫入功能，REST Client 則是將資料發送出去，本範例使用 KEPServer 模擬 tag，透過 IoT Gateway 建立 REST Server 及 Client，並搭配 Node-RED 建立及視覺化資料傳輸流程。

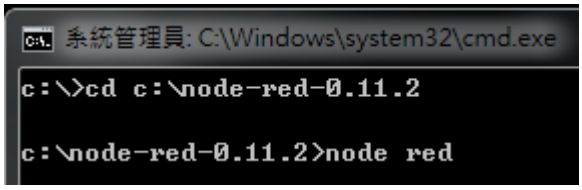
安裝 Node-RED

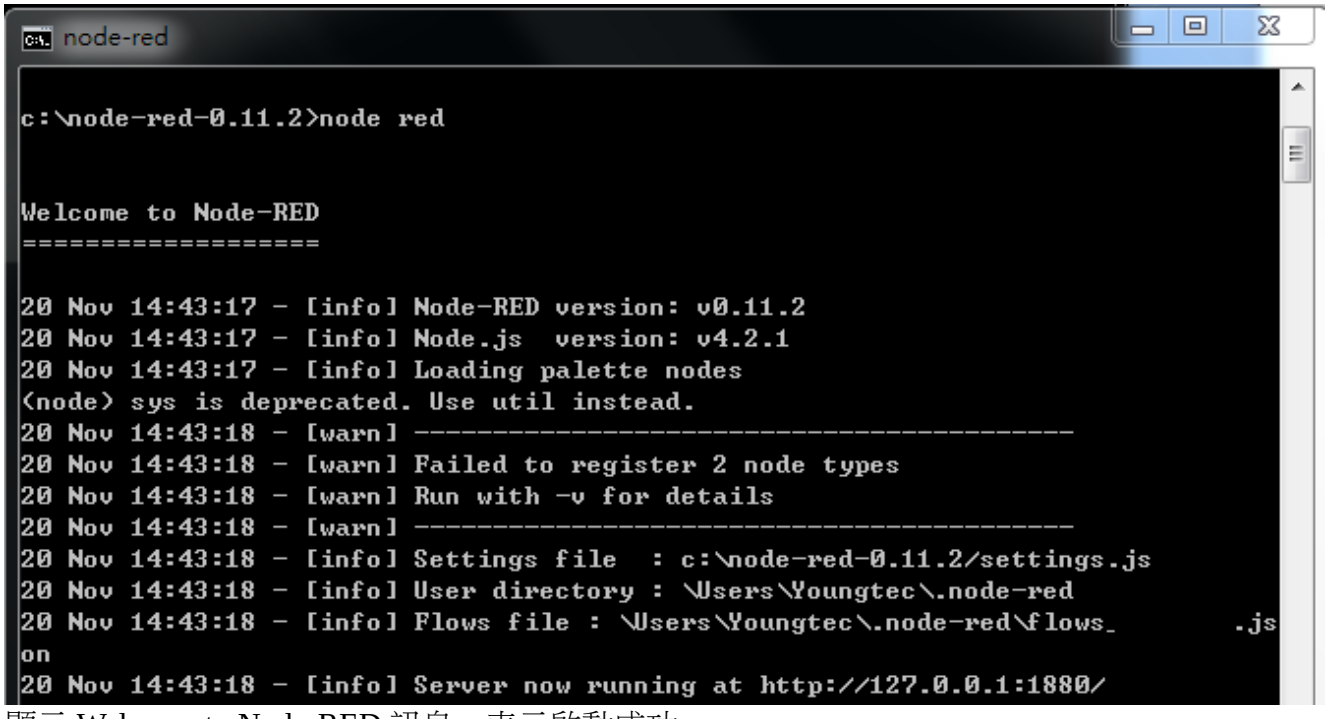
Node-RED 可在瀏覽器上編輯模擬，並視覺化物聯網裝置、APIs 及網路服務連接，我們使用 Node-RED 來呈現 IoT Gateway 的資料傳輸，您需安裝 node.js 及 Node-RED，才可順利執行本範例。

請下載並依指示安裝 node.js，您可至 https://nodejs.org/en/，內有下載及安裝說明。
請下載並依指示安裝 Node-RED，您可至 http://nodered.org/docs/getting-started/installation.html，內有下載及安裝說明。

啟動 Node-RED

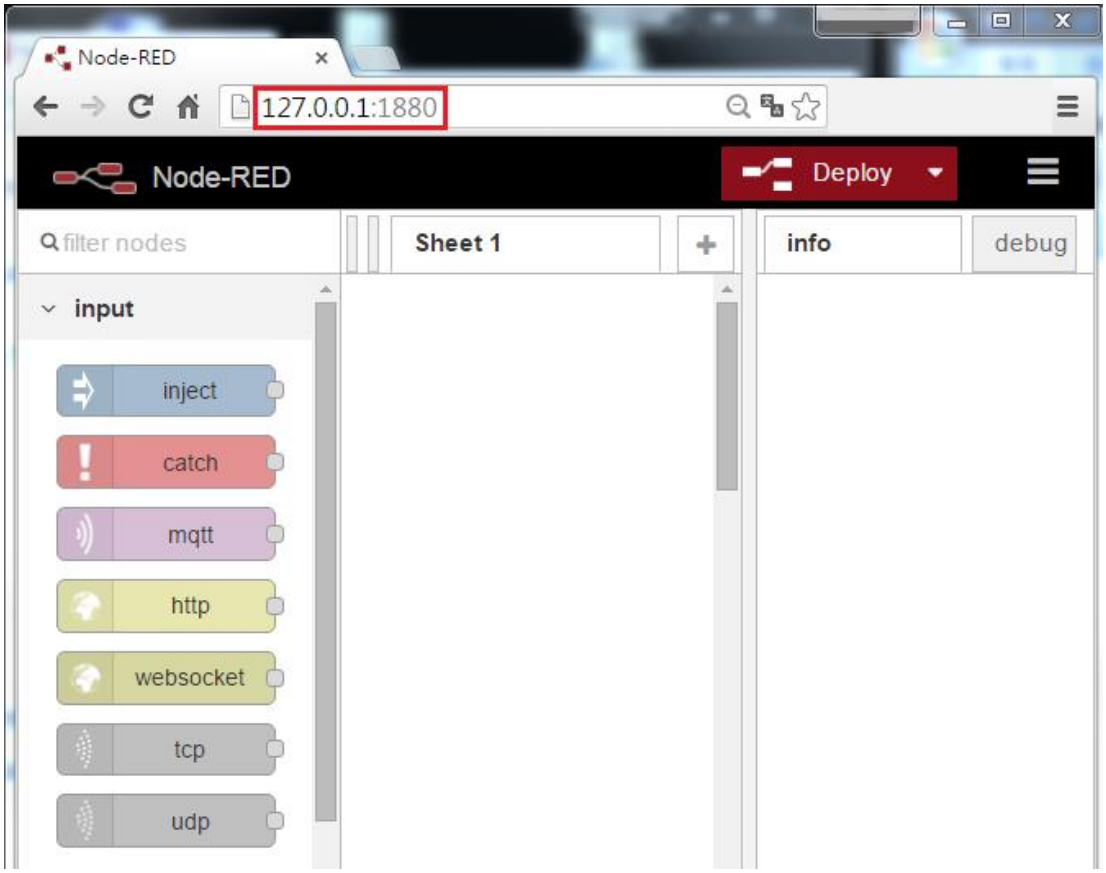
1．開啟命令提示字元(CMD)，切換至 Node-RED 安裝路徑，輸入指令”node red”便可啟動 Node-RED。





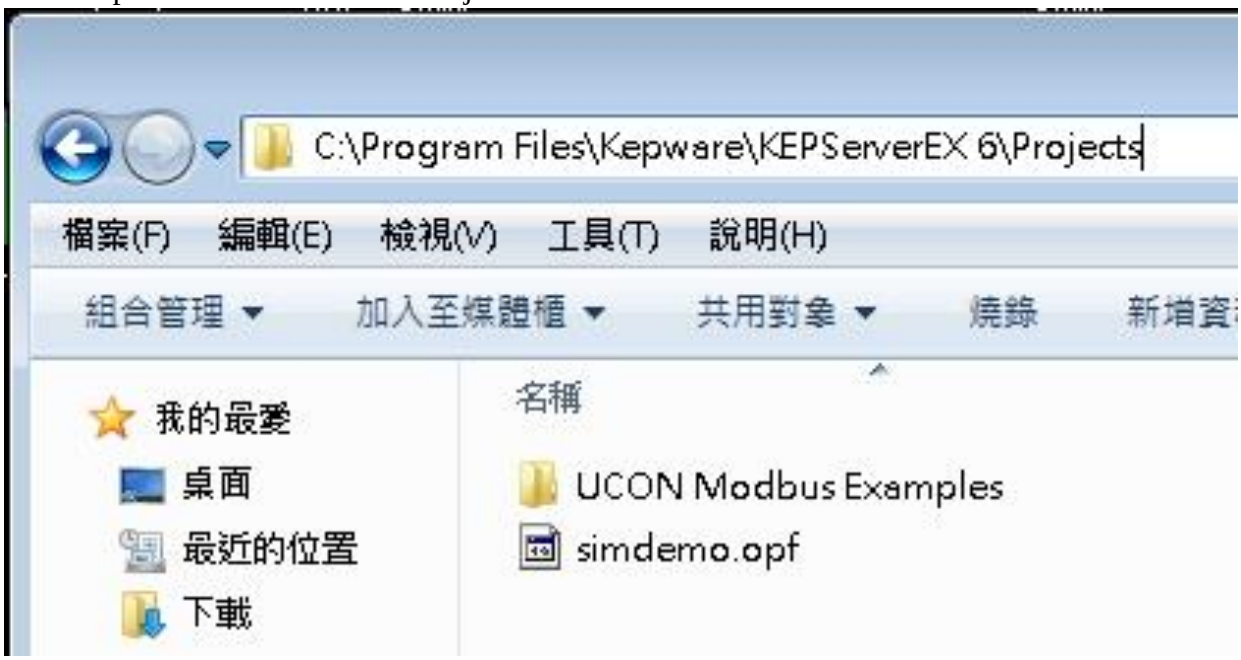
顯示 Welcome to Node-RED 訊息，表示啟動成功。

2．開啟 Chrome 瀏覽器，於網址列輸入 127.0.0.1:1880，即可進入 Node-RED 操作畫面。

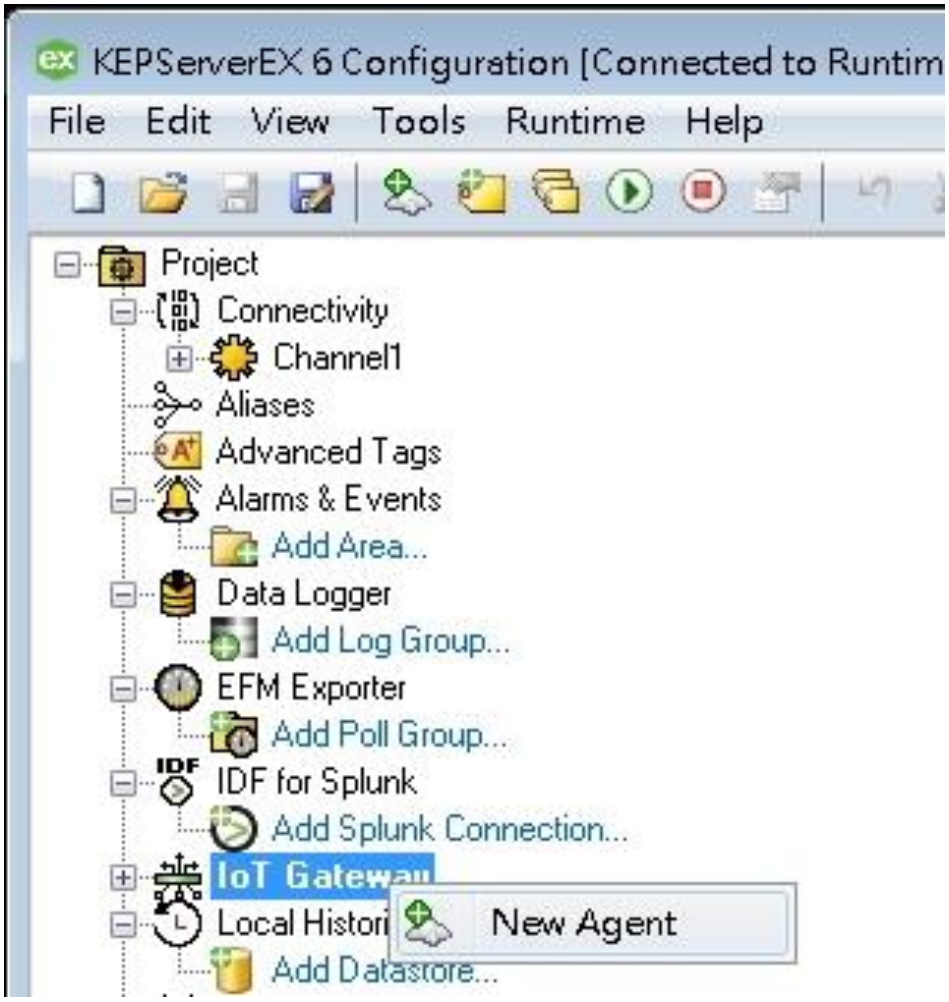


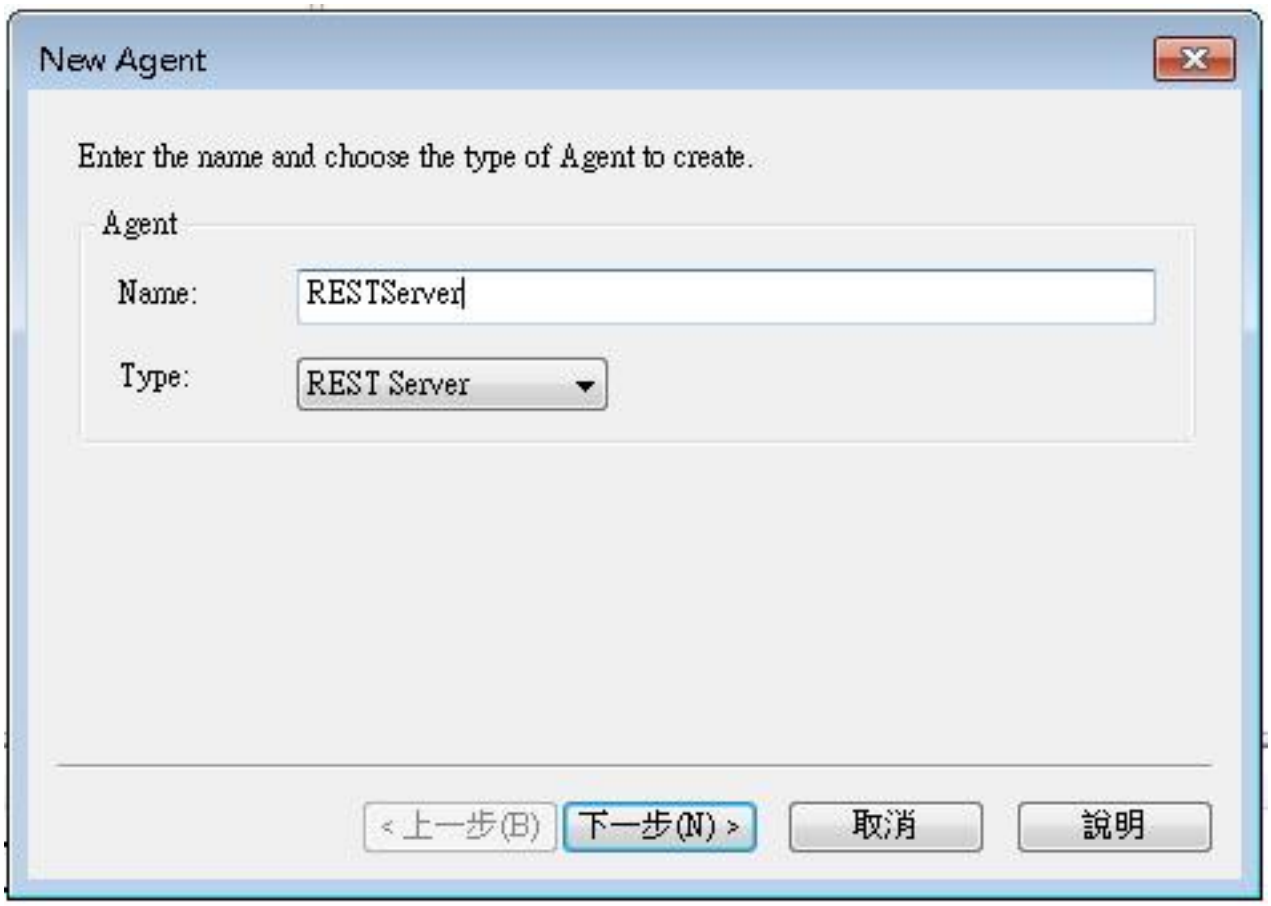
建立 REST Server

1．至 KEPServerEX 安裝路徑，開啟內建的模擬設定檔 simdemo.opf，預設路徑為 C:\Program Files\Kepware\KEPServerEX 6\Projects。

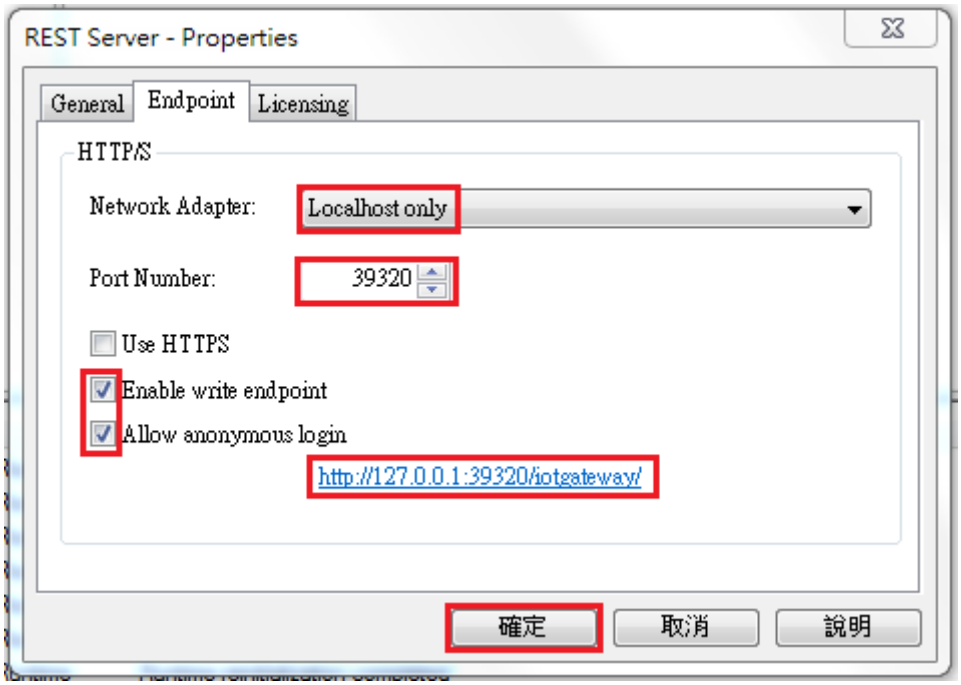


2．在左方樹狀選單的 IoT Gateway 項目，點擊滑鼠右鍵，點擊 New Agent，會顯示 New Agent 視窗，Name 輸入 RESTServer，Type 選擇 REST Server，單擊”下一步”按鈕。

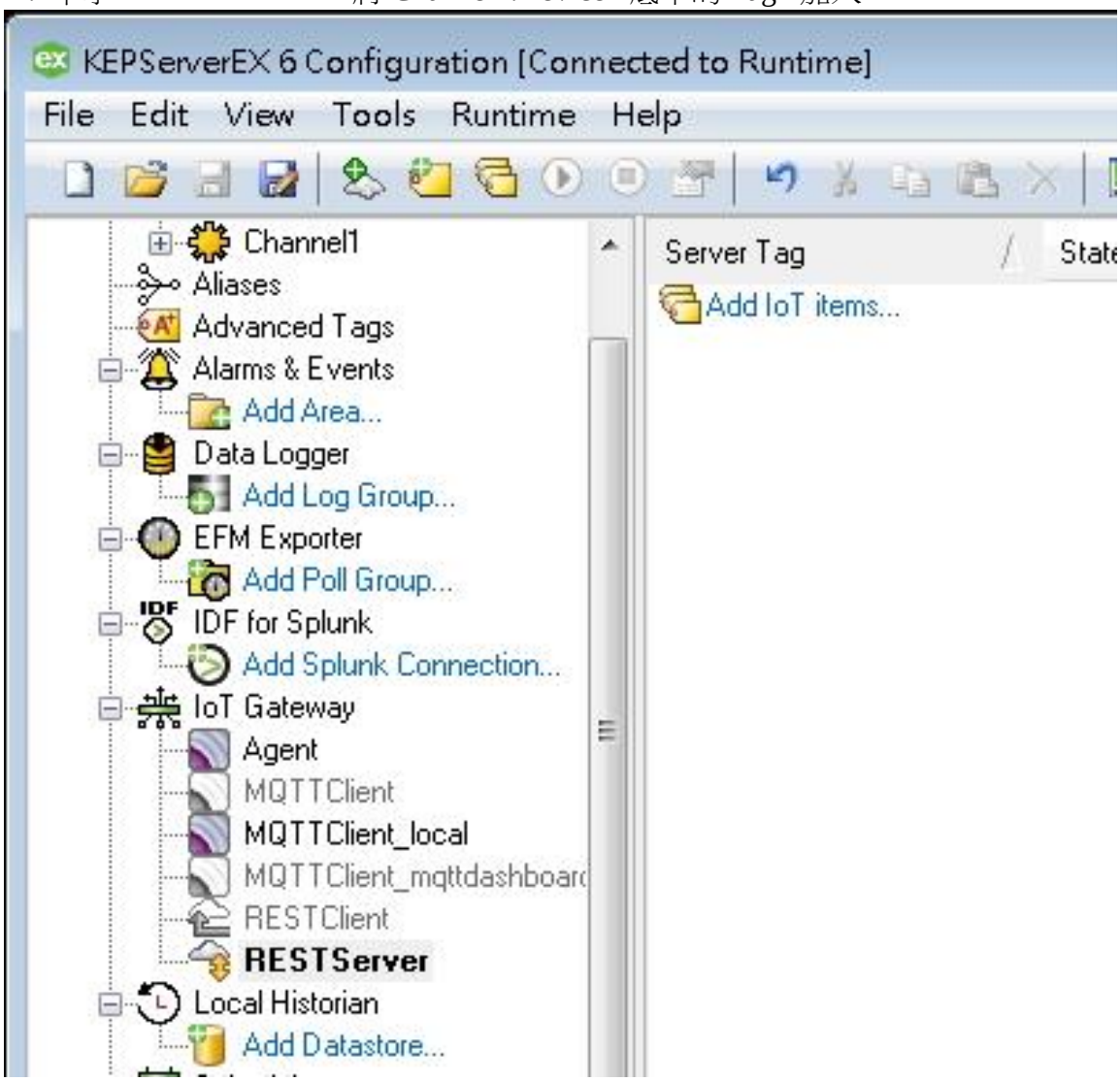


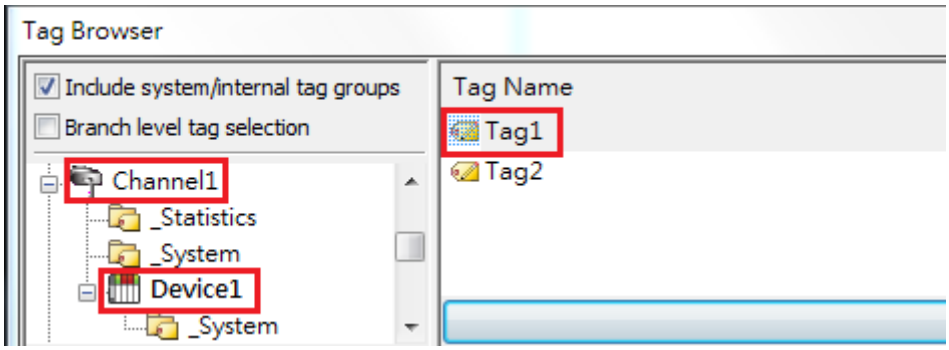


3．設定 Endpoint，Network Adapter 選擇 Localhost only，Port Number 輸入 39320，勾選 Enable write endpoint、Allow anonymous login，底下會顯示 REST Server 的資訊頁面連結，該連結會依照設定而變動。單擊”確定”按鈕。



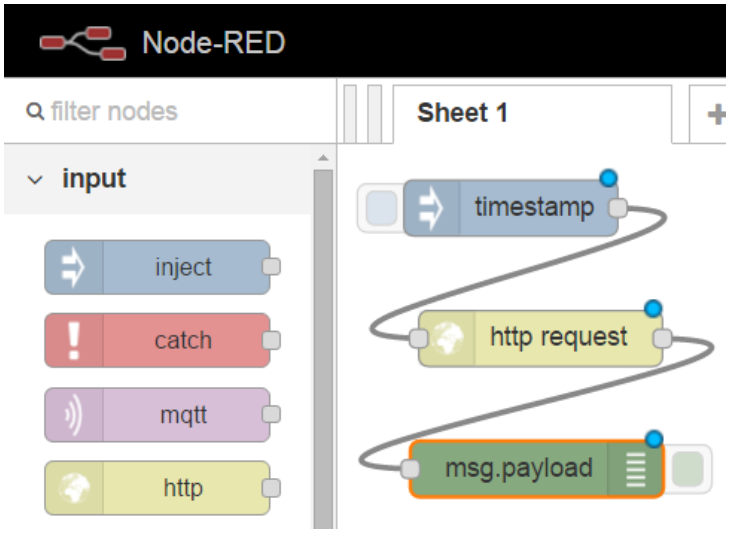
4．單擊”Add IoT items”，將 Channel1.Device1 底下的 Tag1 加入。



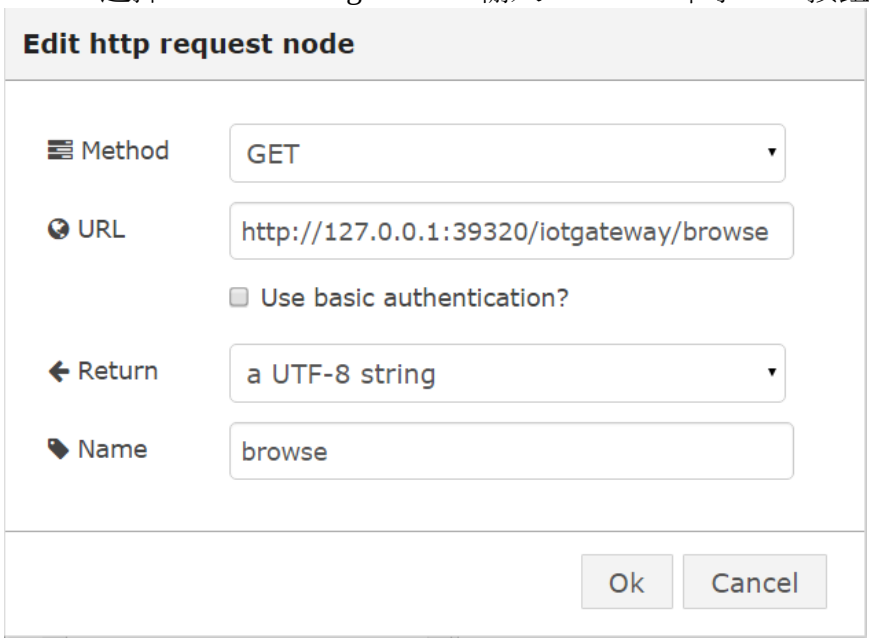


使用 Node-RED 瀏覽 REST Server 的 tag

1．開啟 Node-RED 操作頁面，建立 input 裡的 inject，function 裡的 http request，output 裡的 debug，並連接起來，如下圖。



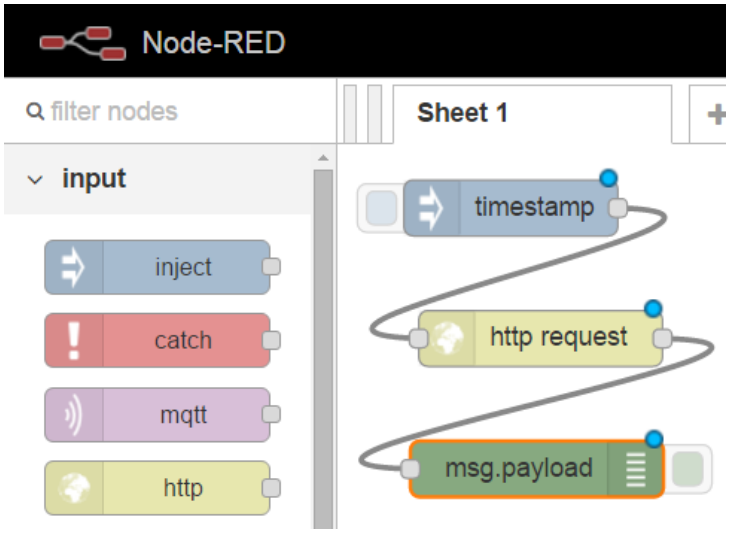
2．編輯 http request 結點，Method 選擇 GET，URL 輸入 http://127.0.0.1:39320/iotgateway/browse，Return 選擇 a UTF-8 string，Name 輸入 browse，單擊”Ok”按鈕。



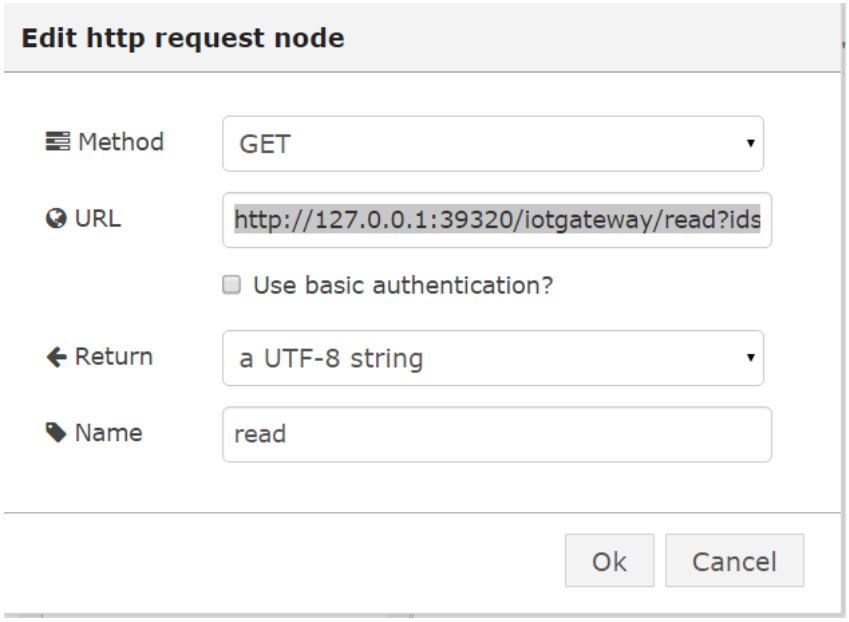
3．單擊右上方”Deploy”按鈕，單擊 timestamp 左方按鈕，便可在右方”debug”頁籤中看到 REST Server 目前可供瀏覽的 tag。

使用 Node-RED 讀取 REST Server 的 tag 資料

1．開啟 Node-RED 操作頁面，建立 input 裡的 inject，function 裡的 http request，output 裡的 debug，並連接起來，如下圖。



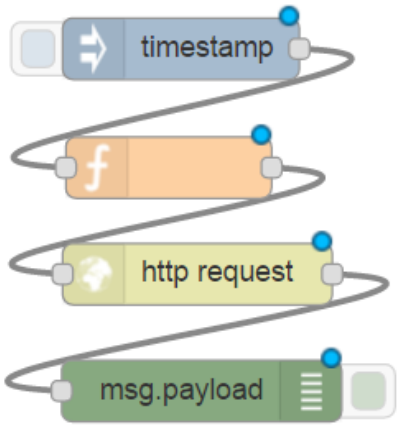
2．編輯 http request 結點，Method 選擇 GET，URL 輸入 http://127.0.0.1:39320/iotgateway/read?ids=Channel1.Device1.tag1，Return 選擇 a UTF-8 string，Name 輸入 read，單擊”Ok”按鈕。



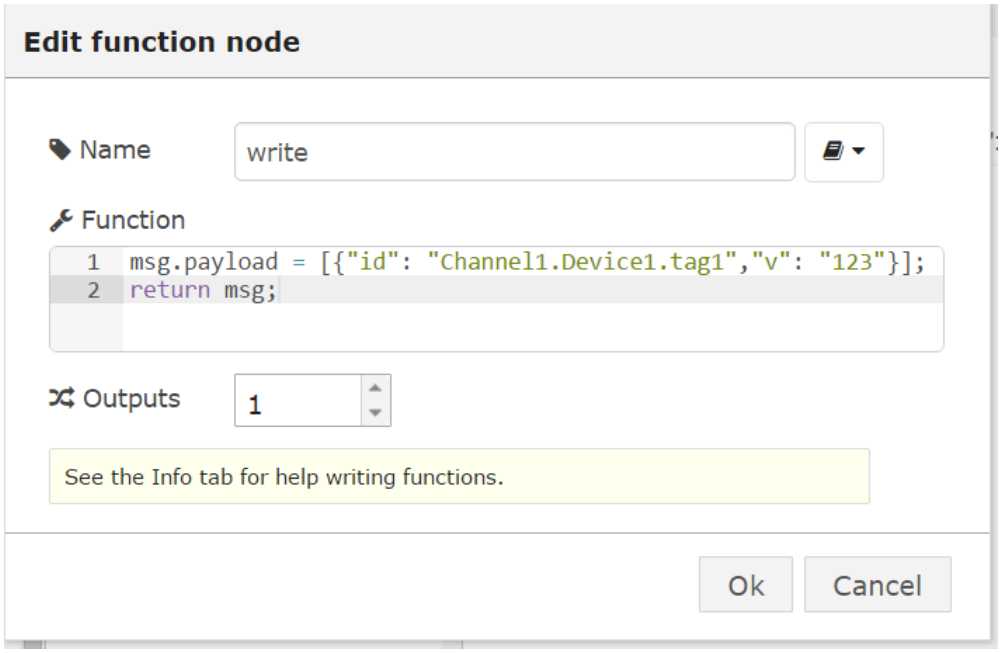
3．單擊右上方”Deploy”按鈕，單擊 timestamp 左方按鈕，便可在右方”debug”頁籤中看到 REST Server 上，Channel1.Device1.tag1 的資料，包括 ID、value、timestamp 等等。

使用 Node-RED 寫入 REST Server 的 tag 數值

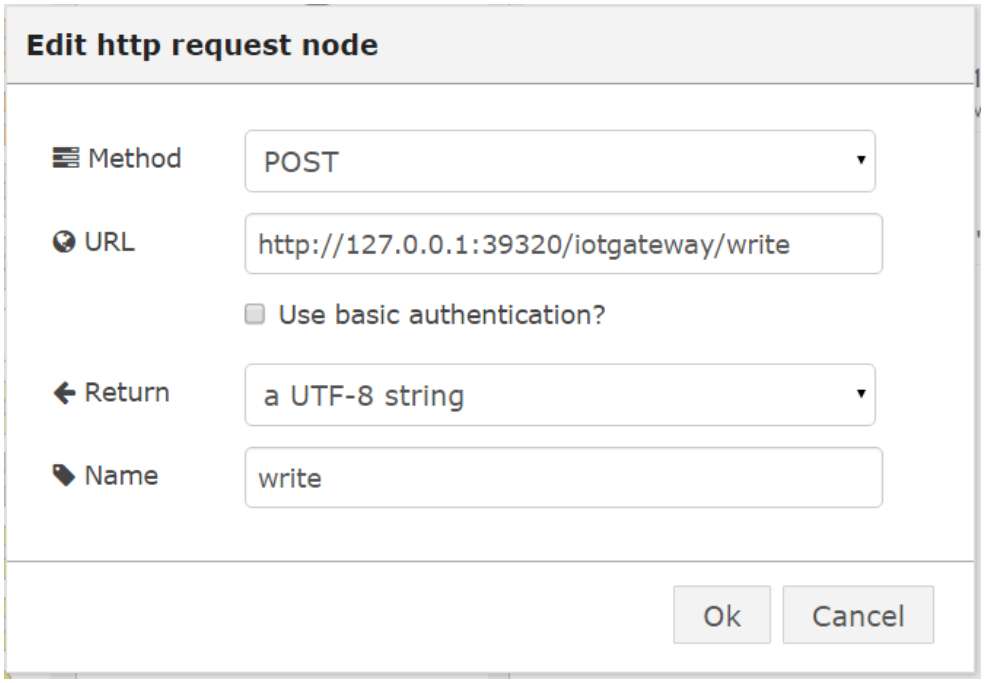
1．開啟 Node-RED 操作頁面，建立 input 裡的 inject，function 裡的 function、http request，output 裡的 debug，並連接起來，如下圖。



2．編輯 function 結點，Name 輸入 write，Function 輸入
msg.payload = [{"id": "Channel1.Device1.tag1","v": "123"}];
return msg;
，單擊”Ok”按鈕。

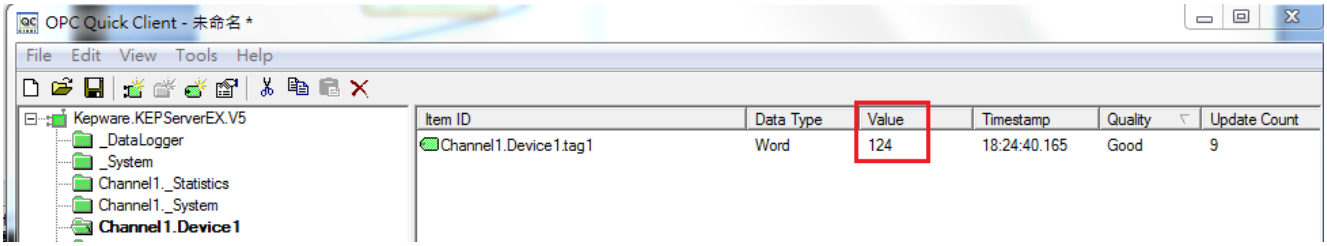


3．編輯 http request 結點，Method 選擇 POST，URL 輸入 http://127.0.0.1:39320/iotgateway/write，Return 選擇 a UTF-8 string，Name 輸入 write，單擊”Ok”按鈕。



4．單擊右上方”Deploy”按鈕，單擊 timestamp 左方按鈕，便可在右方”debug”頁籤中看到寫入成功的訊息。

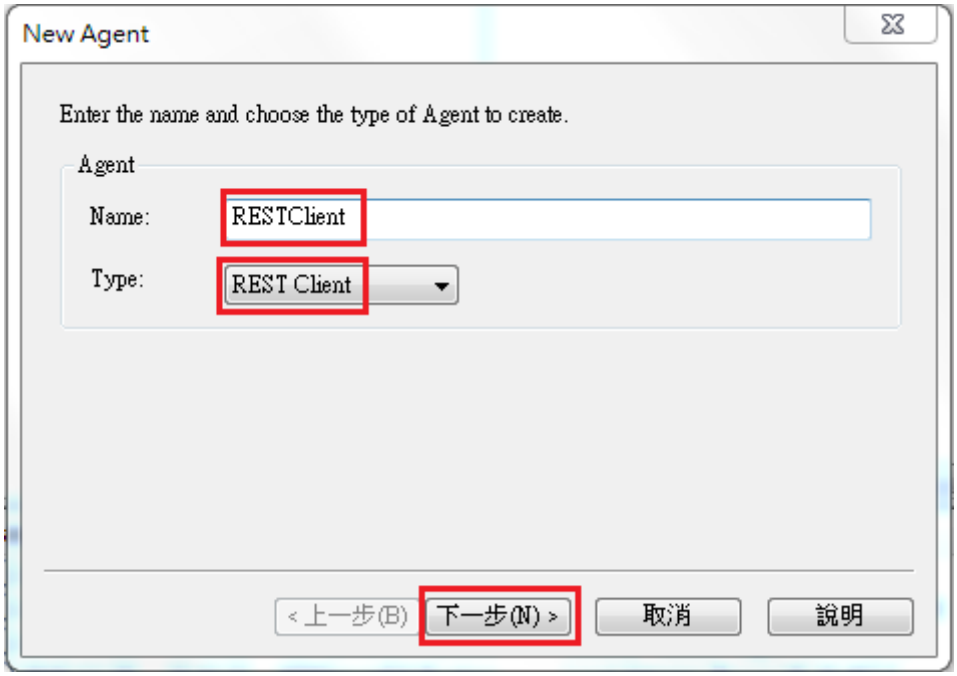
5．開啟 KEPServer 的 Quick Client，可看到 Channel1.Device1.tag1 的數值已被寫入。



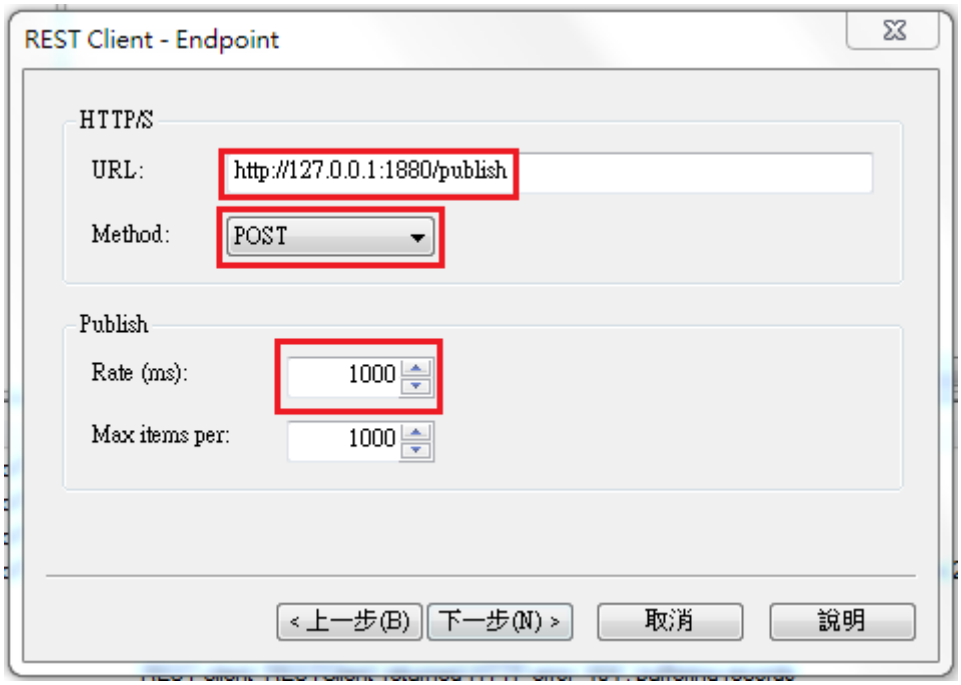
使用 Node-RED 接收 REST Client 發送的資料

1．建立 REST Client。

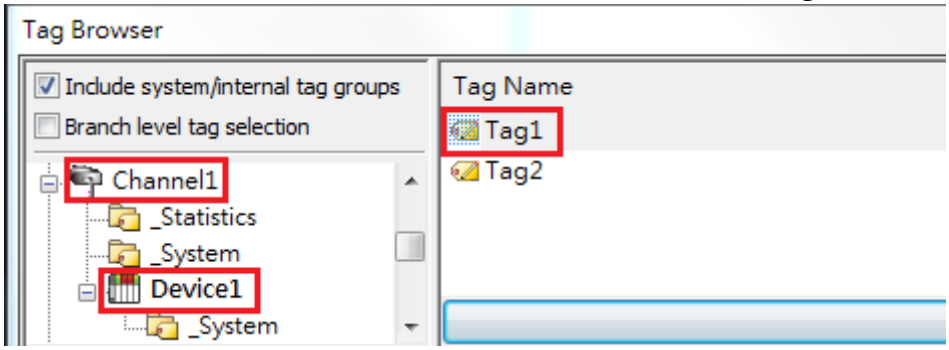
於樹狀選單的 IoT Gateway 項目，點擊滑鼠右鍵，並點擊”New Agent”，會顯示 New Agent 視窗，Name 輸入 RESTClient，Type 選擇 REST Client，單擊”下一步”按鈕。



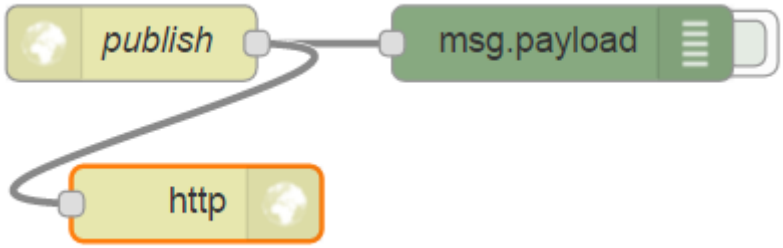
2 . URL 輸入 http://127.0.0.1:1880/publish，Method 選擇 POST，Rate 輸入 1000，單擊”下一步”按鈕，單擊”完成”按鈕。



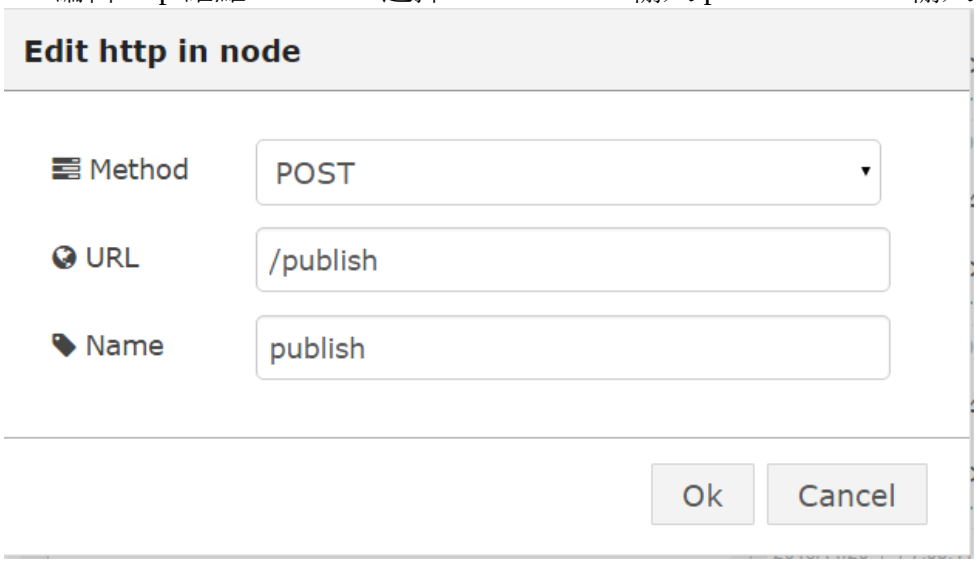
3. 單擊”Add IoT items”， 將 Channel1.Device1 底下的 Tag1 加入。



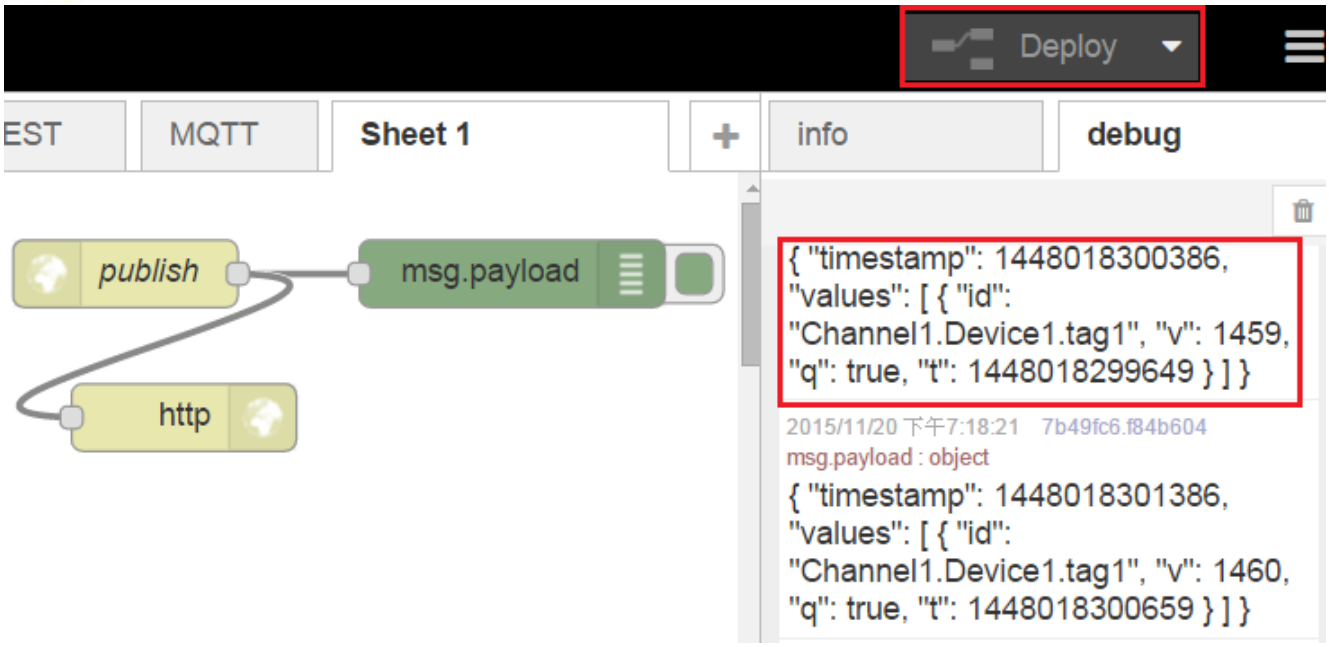
4. 開啟 Node-RED 操作頁面，建立 input 裡的 http， output 裡的 debug、http response，並連接起來，如下圖。



5. 編輯 http 結點，Method 選擇 POST，URL 輸入/publish，Name 輸入 publish，單擊”Ok”按鈕。



6. 單擊右上方的”Deploy”按鈕，便可在右方”debug”標籤頁看到從 KEPServer 中 REST Client 發送的資料。


Deploy
EST
MQTT
Sheet 1
info
debug
publish
msg.payload
http
{ "timestamp": 1448018300386, "values": [{ "id": "Channel1.Device1.tag1", "v": 1459, "q": true, "t": 1448018299649 }] }
2015/11/20 下午7:18:21 7b49fc6.f84b604
msg.payload : object
{ "timestamp": 1448018301386, "values": [{ "id": "Channel1.Device1.tag1", "v": 1460, "q": true, "t": 1448018300659 }] }

MQTT Client 使用範例

IoT Gateway 提供專為物聯網所設計的 MQTT 協定，僅需極少的網路頻寬及硬體資源便可運作，KEPServer 可作為 MQTT 的 Publisher 角色，對 Topic 發送訊息。本範例使用 mosquitto 的 Subscriber 接收 MQTT Client 所發送的訊息並呈現，您需要安裝 mosquitto、OpenSSL 的 libeay32.dll、以及 pthreadVC2.dll。

安裝 mosquitto

請下載並依指示安裝 mosquitto，您可至 http://mosquitto.org/，內有下載及詳細資訊。

安裝 OpenSSL

請下載並依指示安裝 openssl，您可至 https://code.google.com/p/openssl-for-windows/downloads/list 或 https://slproweb.com/products/Win32OpenSSL.html，內有下載及詳細資訊。

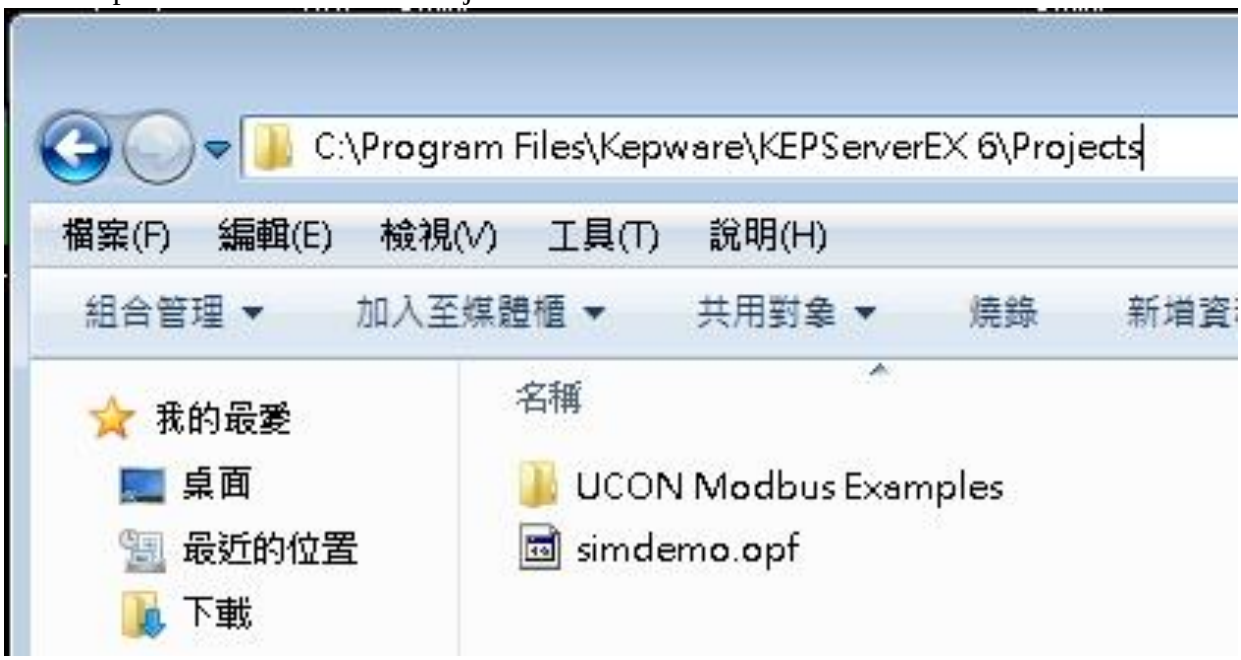
下載 pthreadVC2.dll

請下載 pthreadVC2.dll 檔案，您可至 ftp://sources.redhat.com/pub/pthreads-win32/dll-latest/dll/x86/，內有下載資訊。

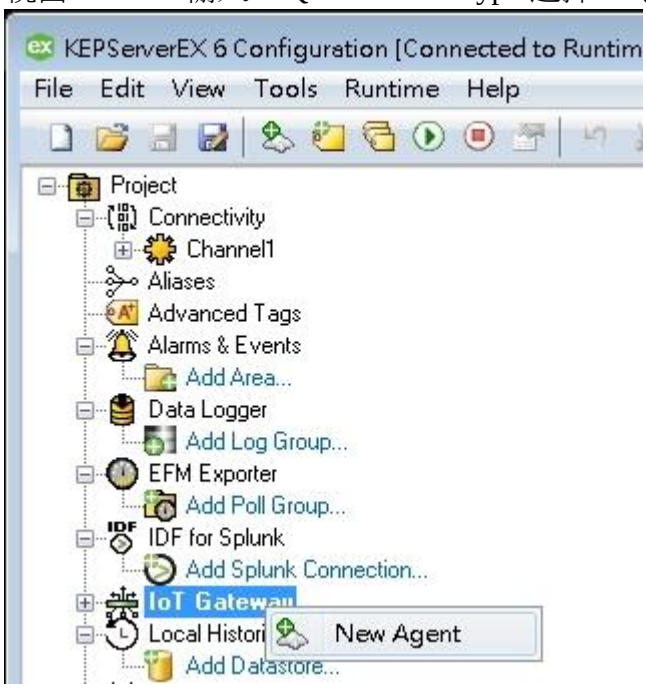
將 pthreadVC2.dll 以及 OpenSSL 安裝路徑底下的 libeay32.dll，複製至 mosquitto 安裝路徑底下。

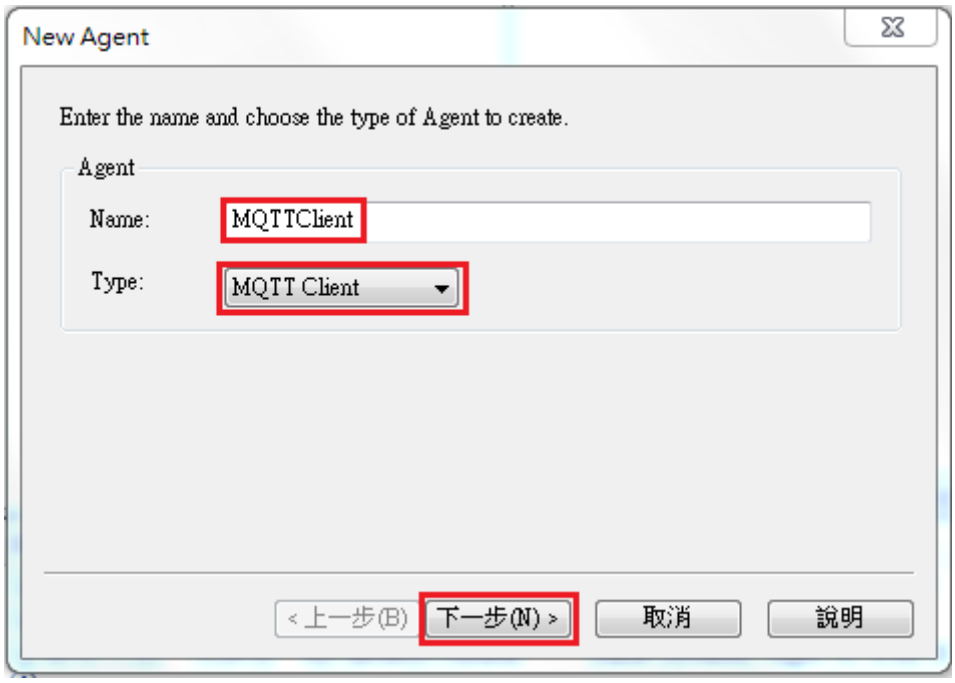
建立 MQTT Client

1．至 KEPServerEX 安裝路徑，開啟內建的模擬設定檔 simdemo.opf，預設路徑為 C:\Program Files\Kepware\KEPServerEX 6\Projects。

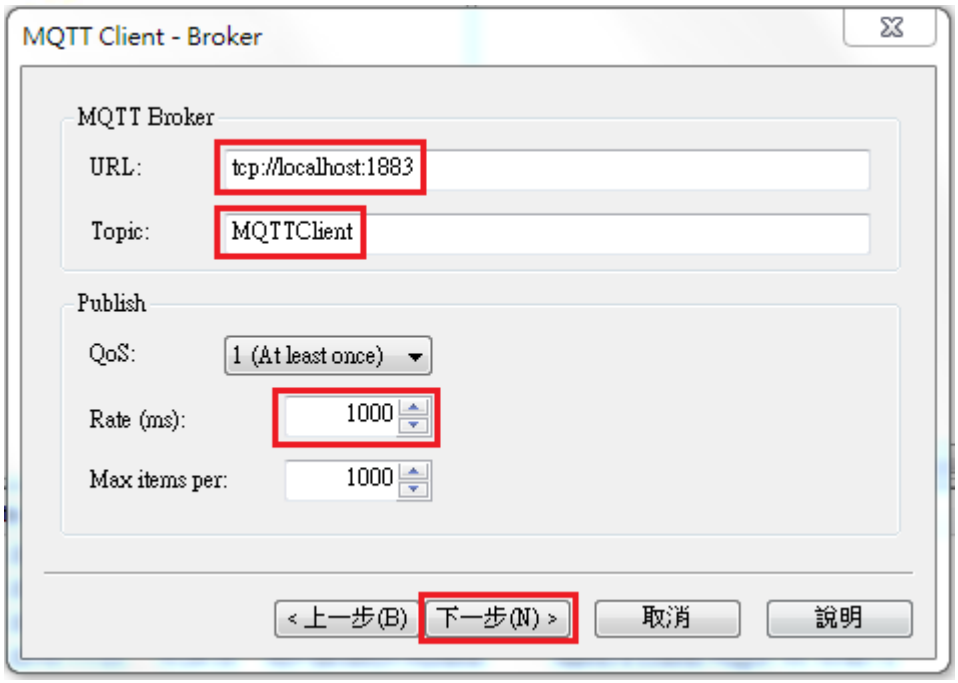


2．在左方樹狀選單的 IoT Gateway 項目，點擊滑鼠右鍵，點擊 New Agent，會顯示 New Agent 視窗，Name 輸入 MQTTClient，Type 選擇 MQTT Client，單擊”下一步”按鈕。

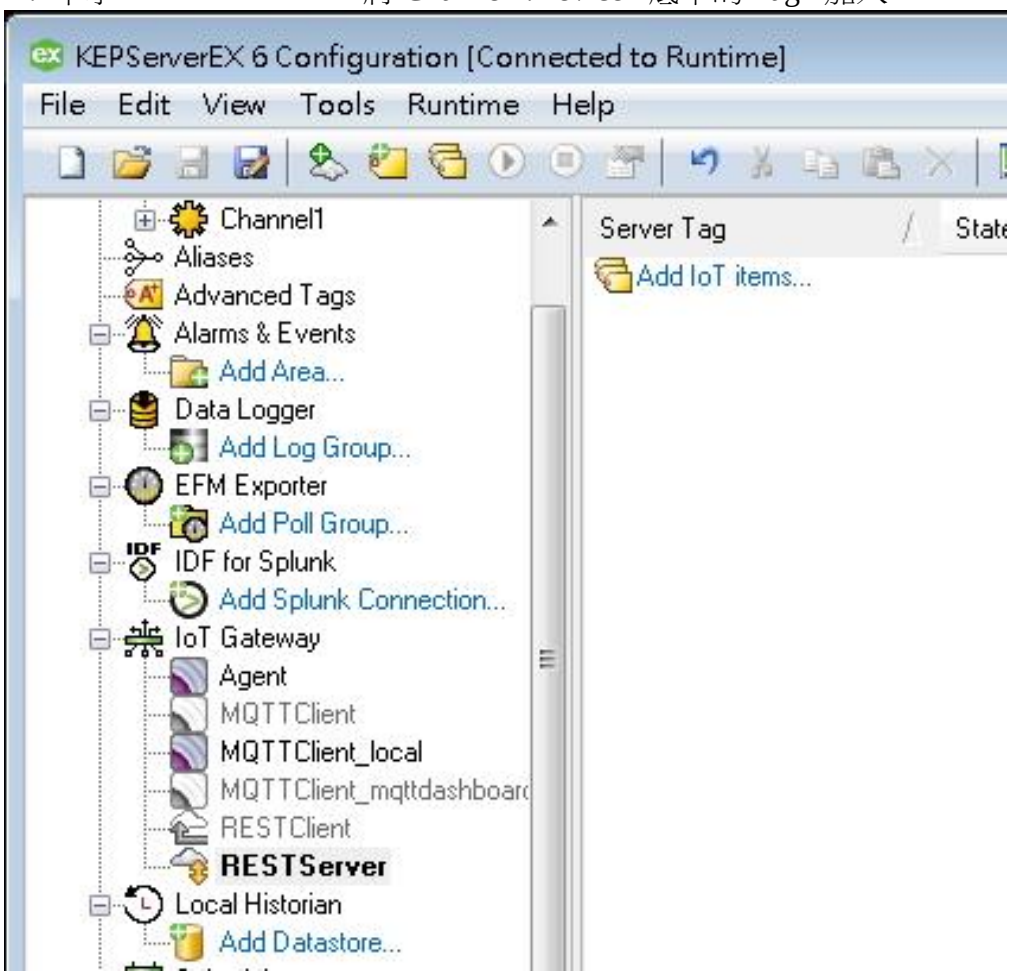


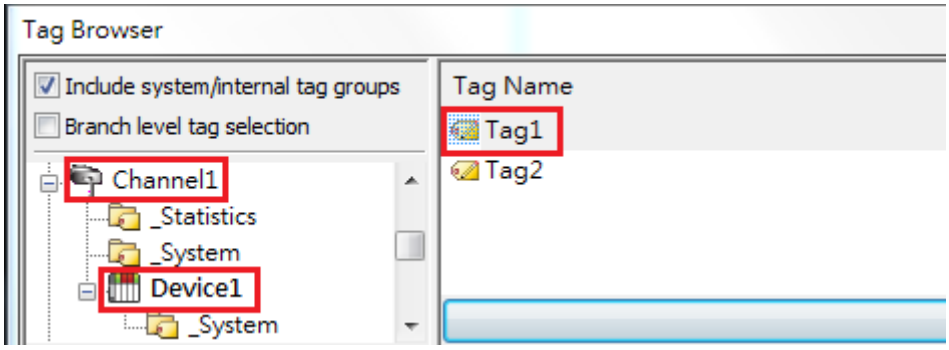


3．於 URL 輸入 tcp://localhost:1883，於 Topic 輸入 MQTTClient，於 Rate(ms)輸入 1000，單擊”下一步”按鈕，單擊”完成”按鈕。


MQTT Client - Broker
MQTT Broker
URL:
tcp://localhost:1883
Topic:
MQTTClient
Publish
QoS:
1 (At least once)
Rate (ms):
1000
Max items per:
1000
< 上一步(B)
下一步(N) >
取消
說明

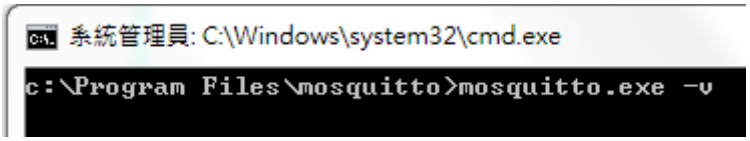
4．單擊”Add IoT items”，將 Channel1.Device1 底下的 Tag1 加入。



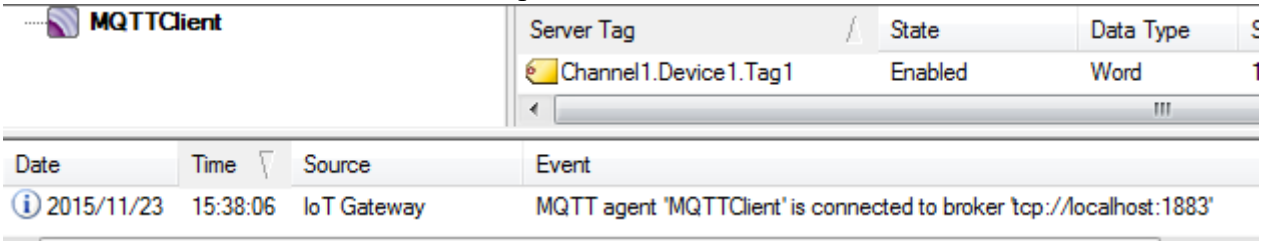


開啟 mosquitto broker

1 .開啟命令提示字元(CMD)，切換路徑至 mosquitto 安裝路徑，輸入指令”mosquito.exe -v”便可啟動 broker。

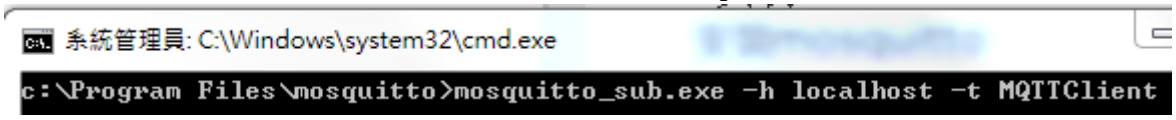


2．啟動 broker 後，KEPServer Configuration 底下的 event 訊息便出現”MQTT agent ‘MQTTClient’is connected to broker ‘tcp://localhost:1883’”，表示 MQTT Client 順利連接至 broker。

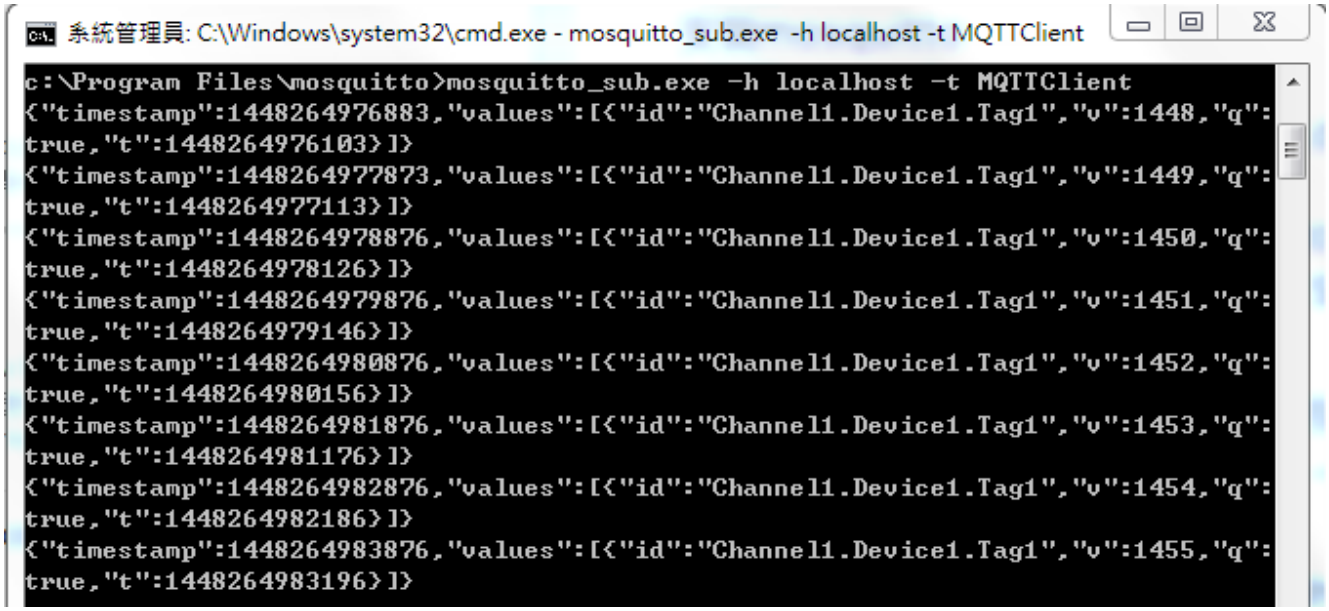


使用 mosquitto Subscriber 接收訊息

1．開啟命令提示字元(CMD)，切換路徑至 mosquitto 安裝路徑，輸入指令” mosquitto_sub.exe -h localhost -t MQTTClient”便可開啟 subscriber 訂閱 Topic 的訊息。



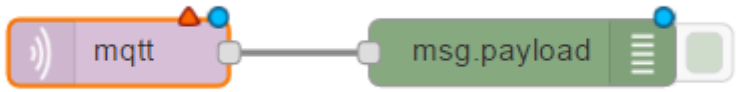
2．可看到由 MQTT Client 發送的訊息，包含 timestamp、values、id 等等。



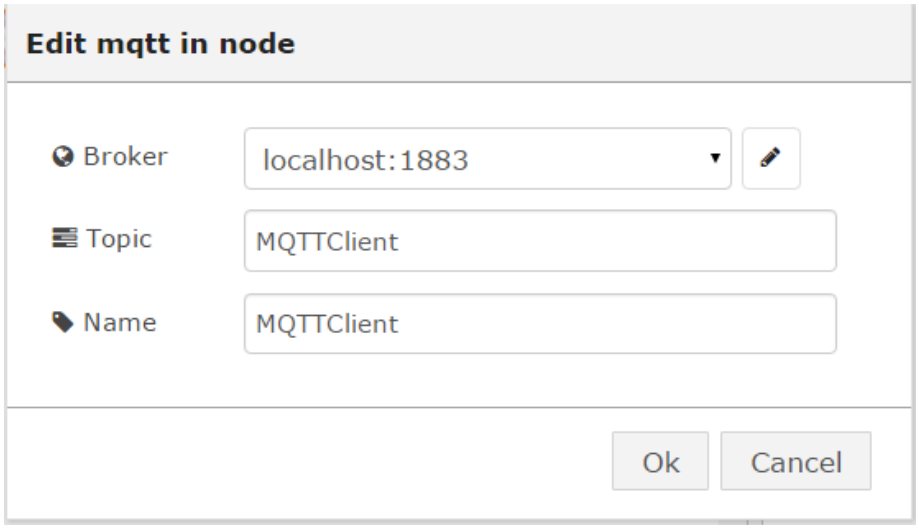
使用 mosquitto Subscriber 接收訊息

您也可使用 Node-RED 來建立並接收 MQTT Client 所發送的訊息。

1．進入 Node-RED 操作畫面後，新增 input 裡的 mqtt，output 裡的 debug，並連接起來。



2．編輯 mqtt 內容，於 Broker 輸入 localhost:1883，Topic 輸入 MQTTClient，Name 輸入 MQTTClient，單擊”Ok”按鈕。



3．單擊右上方”Deploy”按鈕，便可在”debug”頁籤看到 MQTT Client 所發送的訊息。