Injecting to InfluxDB directly from a Compactlogix PLC

Contents

1. HTTP Library

You can see the final result on this video

https://youtu.be/UafslB52IMU

PLC has DHCP address 192.168.1.41 connected thru a patchcord to the router.

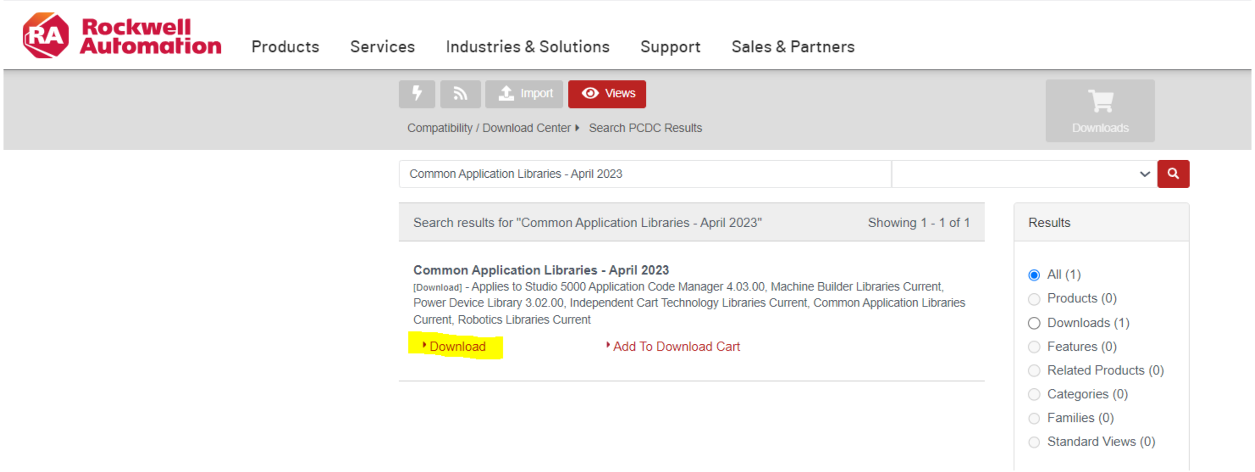
Computer is connected the router thru wifi.

Computer has wifi DHCP address 192.168.1.163

You can find this AOI RM-raC_Opr_HTTP on this library “CommonApplicationLibraries_20230415”

This belongs to Common application library to be download from Rockwell download page.

https://compatibility.rockwellautomation.com/pages/search.aspx?crumb=117&q=Common%20Application%20Libraries%20-%20April%202023



Common Application Libraries - April 2023

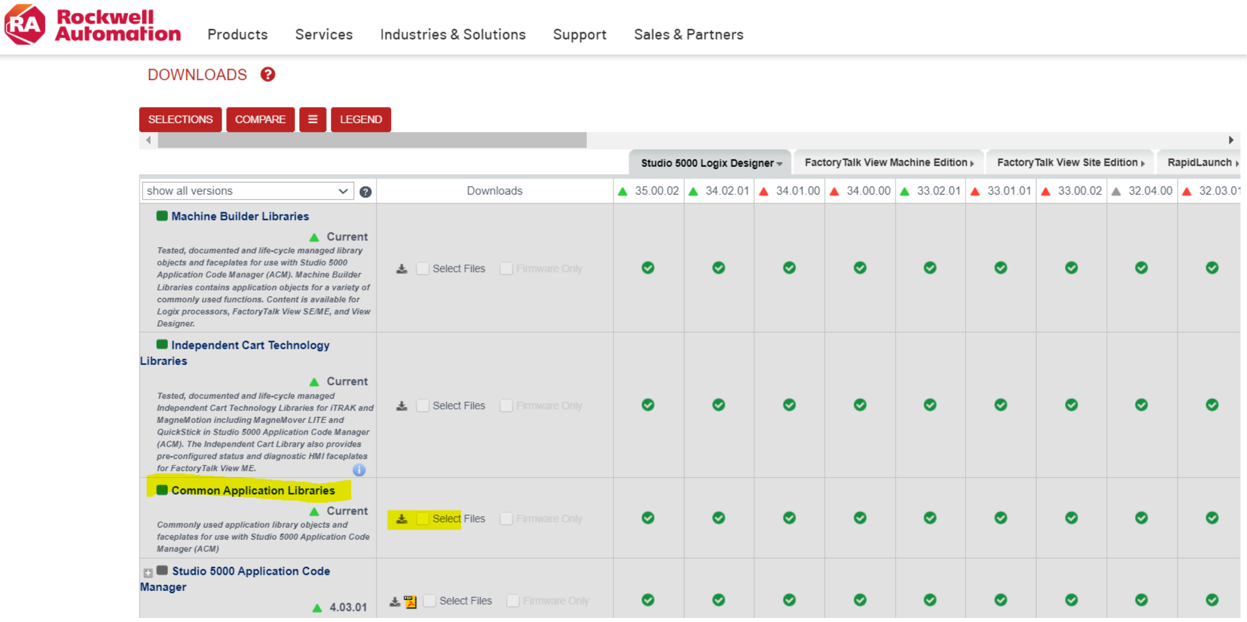
Search results for "Common Application Libraries - April 2023" Showing 1 - 1 of 1

Common Application Libraries - April 2023

[Download] - Applies to Studio 5000 Application Code Manager 4.03.00, Machine Builder Libraries Current, Power Device Library 3.02.00, Independent Cart Technology Libraries Current, Common Application Libraries Current, Robotics Libraries Current

▸ Download ▸ Add To Download Cart

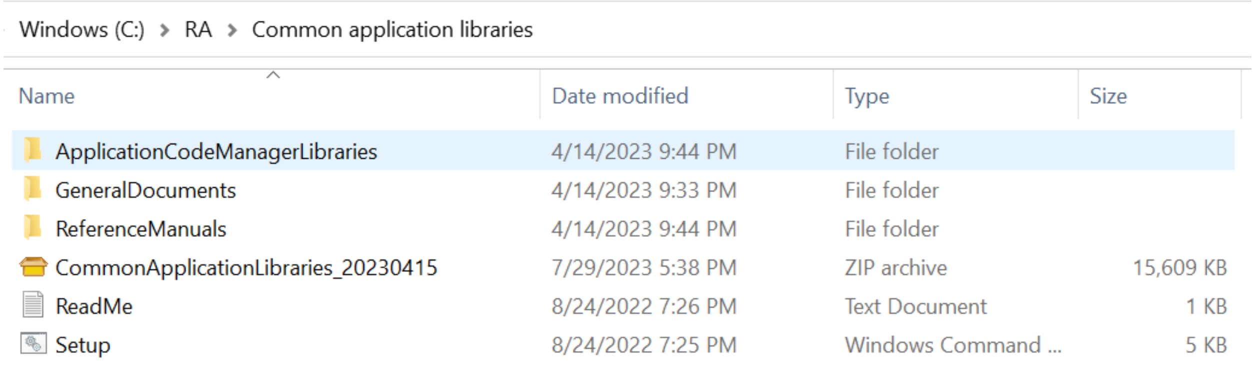
● All (1)
○ Products (0)
○ Downloads (1)
○ Features (0)
○ Related Products (0)
○ Categories (0)
○ Families (0)
○ Standard Views (0)



show all versions	Downloads	▲ 35.00.02	▲ 34.02.01	▲ 34.01.00	▲ 34.00.00	▲ 33.02.01	▲ 33.01.01	▲ 33.00.02	▲ 32.04.00	▲ 32.03.01
Machine Builder Libraries ▲ Current *Tested, documented and life-cycle managed library objects and faceplates for use with Studio 5000 Application Code Manager (ACM). Machine Builder Libraries contains application objects for a variety of commonly used functions. Content is available for Logix processors, FactoryTalk View SE/ME, and View Designer.*	☐ Select Files ☐ Firmware Only	✓	✓	✓	✓	✓	✓	✓	✓	✓
Independent Cart Technology Libraries ▲ Current *Tested, documented and life-cycle managed Independent Cart Technology Libraries for iTRAK and MagneMotion including MagneMover LITE and QuickStick in Studio 5000 Application Code Manager (ACM). The Independent Cart Library also provides pre-configured status and diagnostic HMI faceplates for FactoryTalk View ME.*	☐ Select Files ☐ Firmware Only	✓	✓	✓	✓	✓	✓	✓	✓	✓
Common Application Libraries ▲ Current *Commonly used application library objects and faceplates for use with Studio 5000 Application Code Manager (ACM)*	☐ Select Files ☐ Firmware Only	✓	✓	✓	✓	✓	✓	✓	✓	✓
Studio 5000 Application Code Manager ▲ 4.03.01	☐ Select Files ☐ Firmware Only	✓	✓	✓	✓	✓	✓	✓	✓	✓

You will get these content on C:\RA

Download

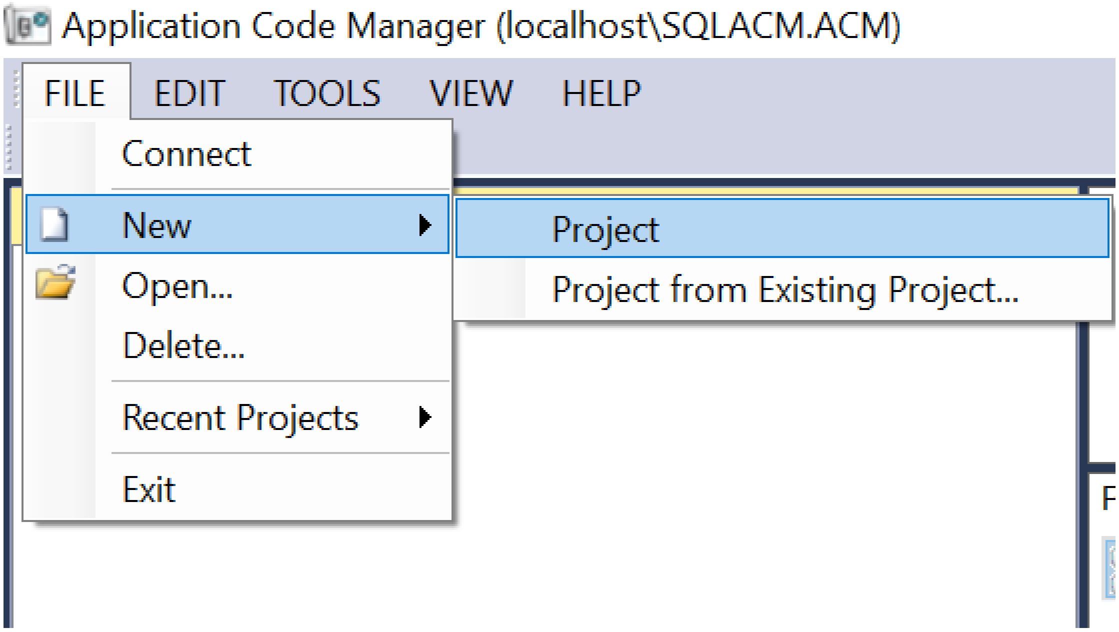


Name	Date modified	Type	Size
ApplicationCodeManagerLibraries	4/14/2023 9:44 PM	File folder	
GeneralDocuments	4/14/2023 9:33 PM	File folder	
ReferenceManuals	4/14/2023 9:44 PM	File folder	
CommonApplicationLibraries_20230415	7/29/2023 5:38 PM	ZIP archive	15,609 KB
ReadMe	8/24/2022 7:26 PM	Text Document	1 KB
Setup	8/24/2022 7:25 PM	Windows Command ...	5 KB

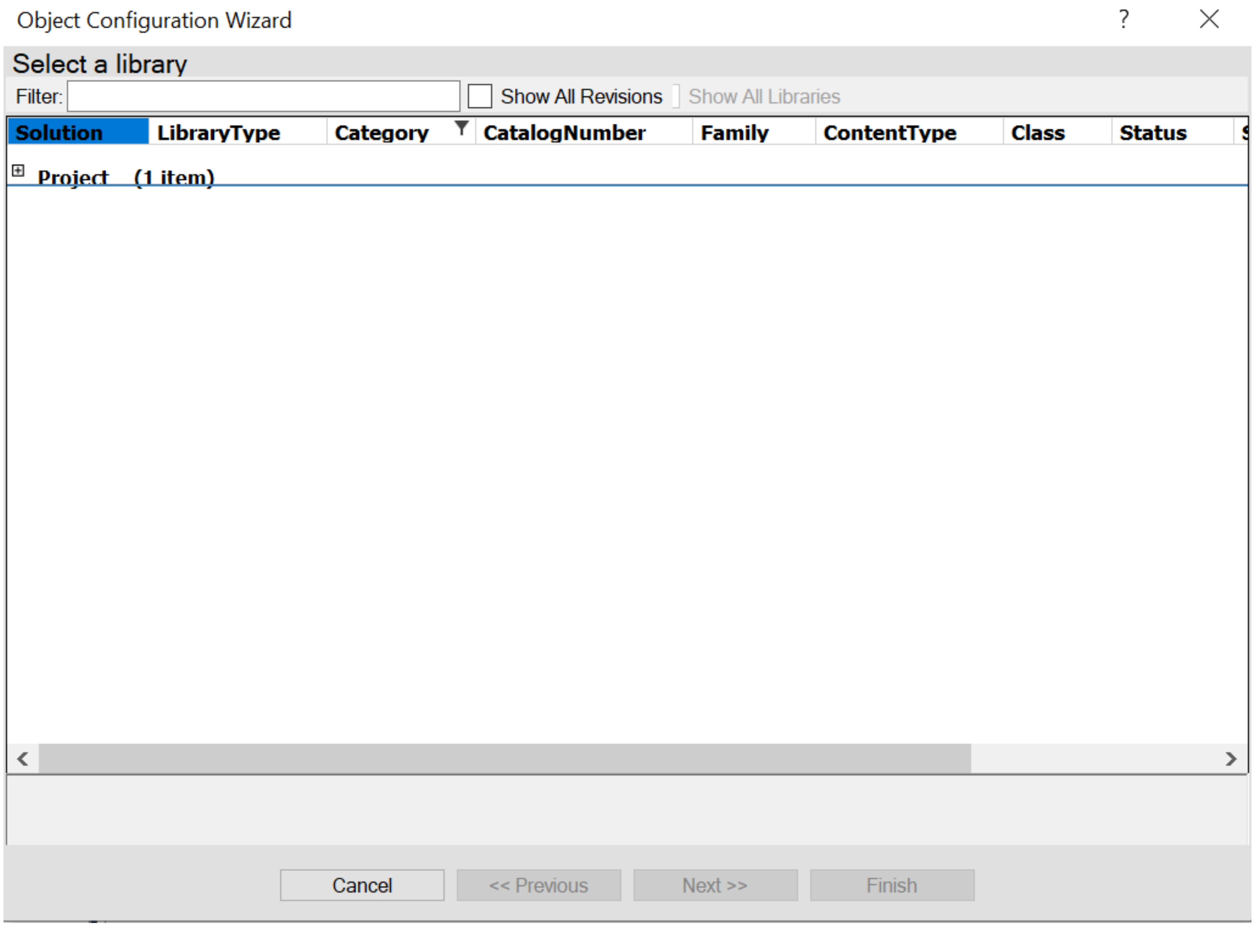
Run Setup in order to install the libraries on ACM

Open ACM

Create a new project


Application Code Manager (localhost\SQLACM.ACM)
FILE
EDIT
TOOLS
VIEW
HELP
Connect
New
Open...
Delete...
Recent Projects
Exit
Project
Project from Existing Project...
F

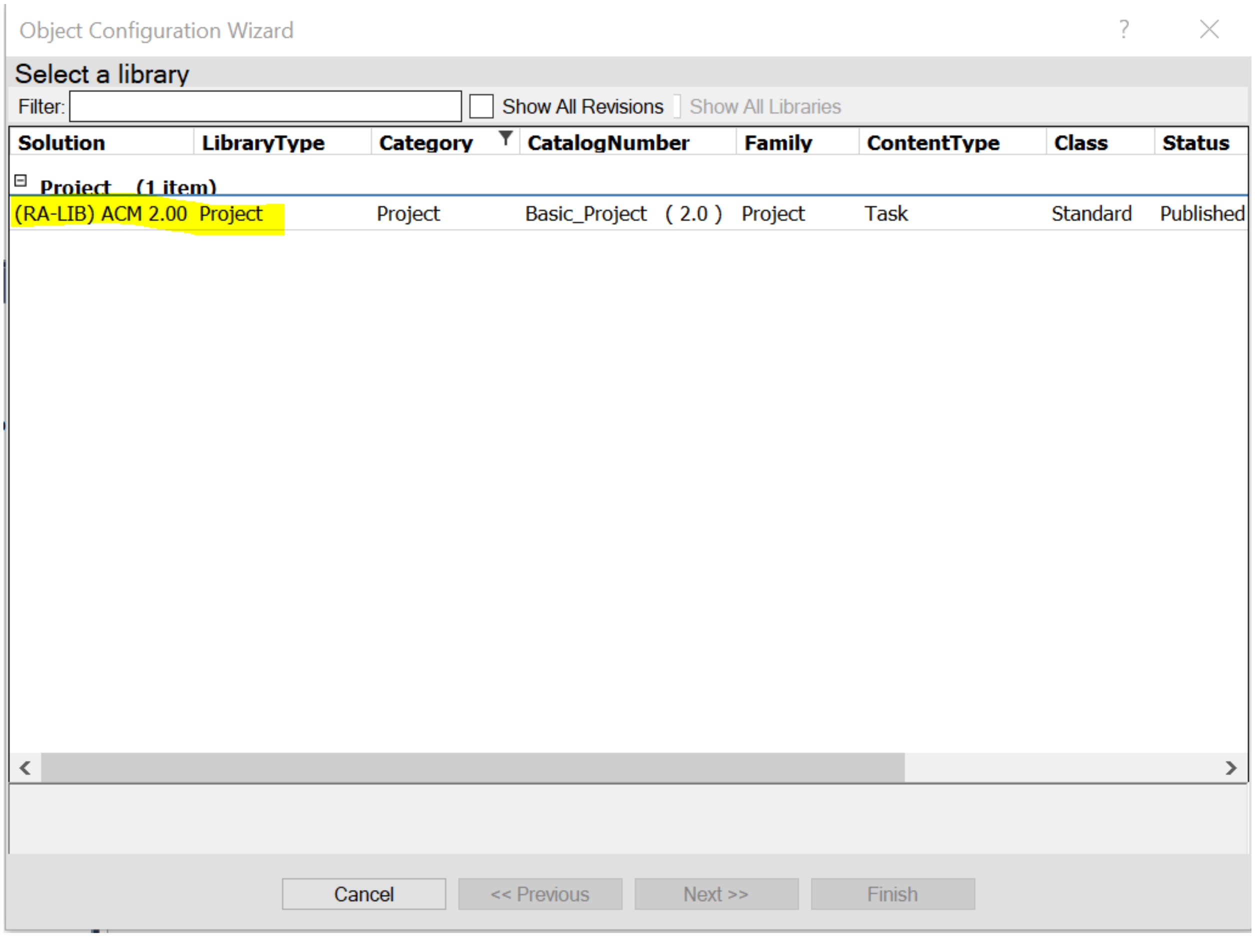
Object configuration Wizard will open



Solution	LibraryType	Category	CatalogNumber	Family	ContentType	Class	Status
⊞ Project (1 item)							

Unfold the Project

Click on the yellow area

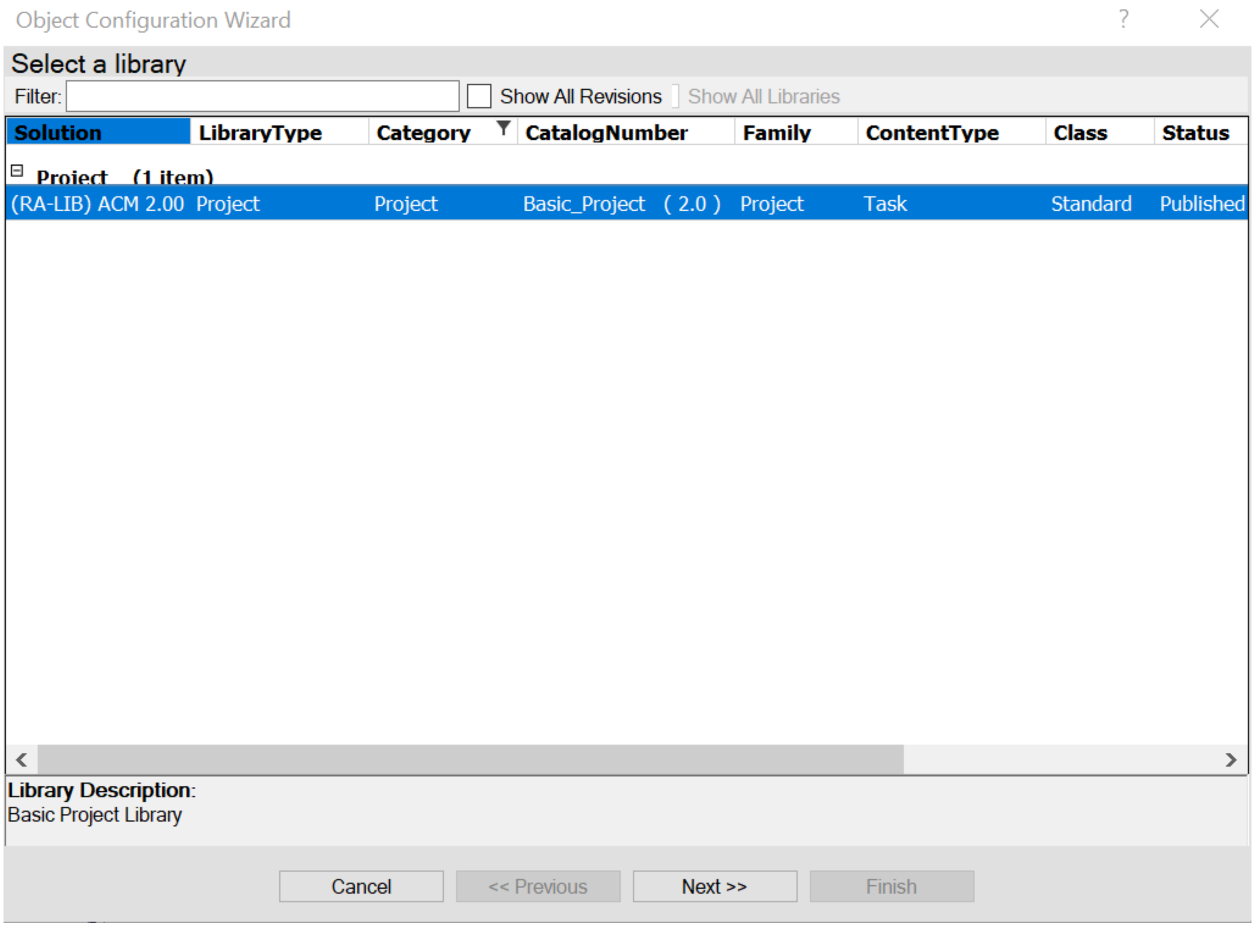


Select a library

Filter: □ Show All Revisions □ Show All Libraries

Solution	LibraryType	Category	CatalogNumber	Family	ContentType	Class	Status
⊟ Project (1 item)							
(RA-LIB) ACM 2.00	Project	Project	Basic_Project (2.0)	Project	Task	Standard	Published

Click on Next

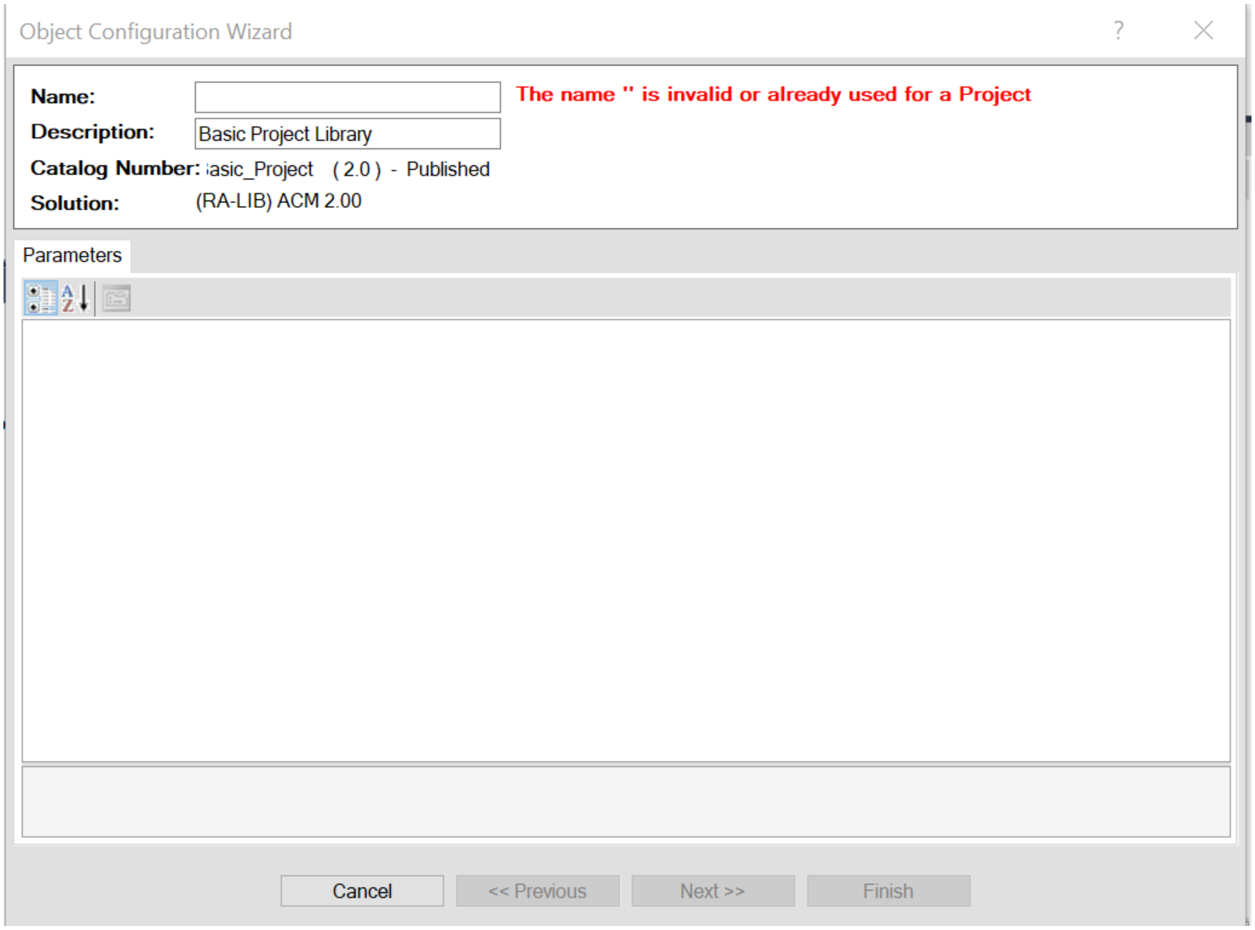


⊟ Project (1 item)

Solution	LibraryType	Category	CatalogNumber	Family	ContentType	Class	Status
(RA-LIB) ACM 2.00	Project	Project	Basic_Project (2.0)	Project	Task	Standard	Published

Library Description:
Basic Project Library

Give a name



Object Configuration Wizard

Name:

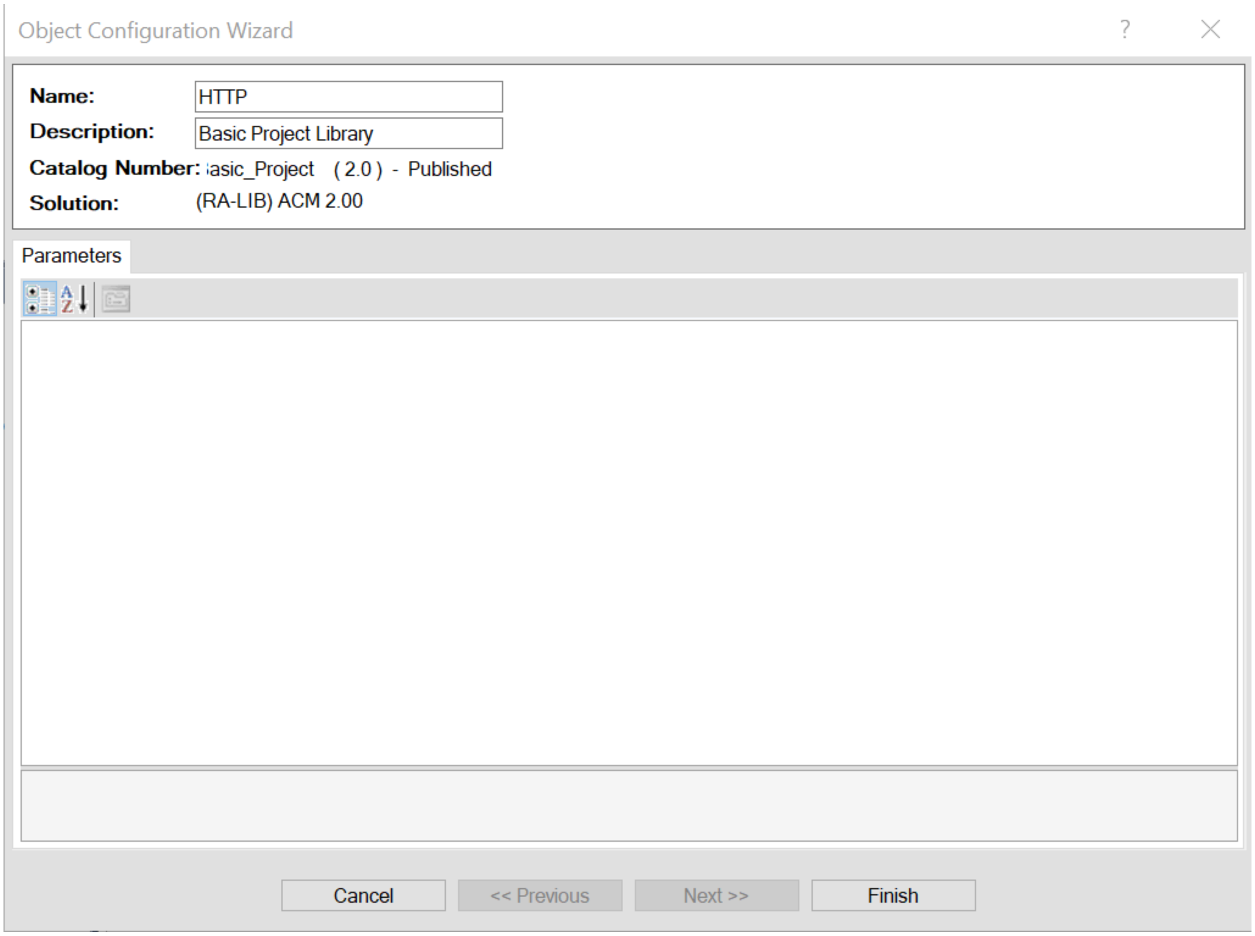
The name '' is invalid or already used for a Project

Description: Basic Project Library

Catalog Number: ›asic_Project (2.0) - Published

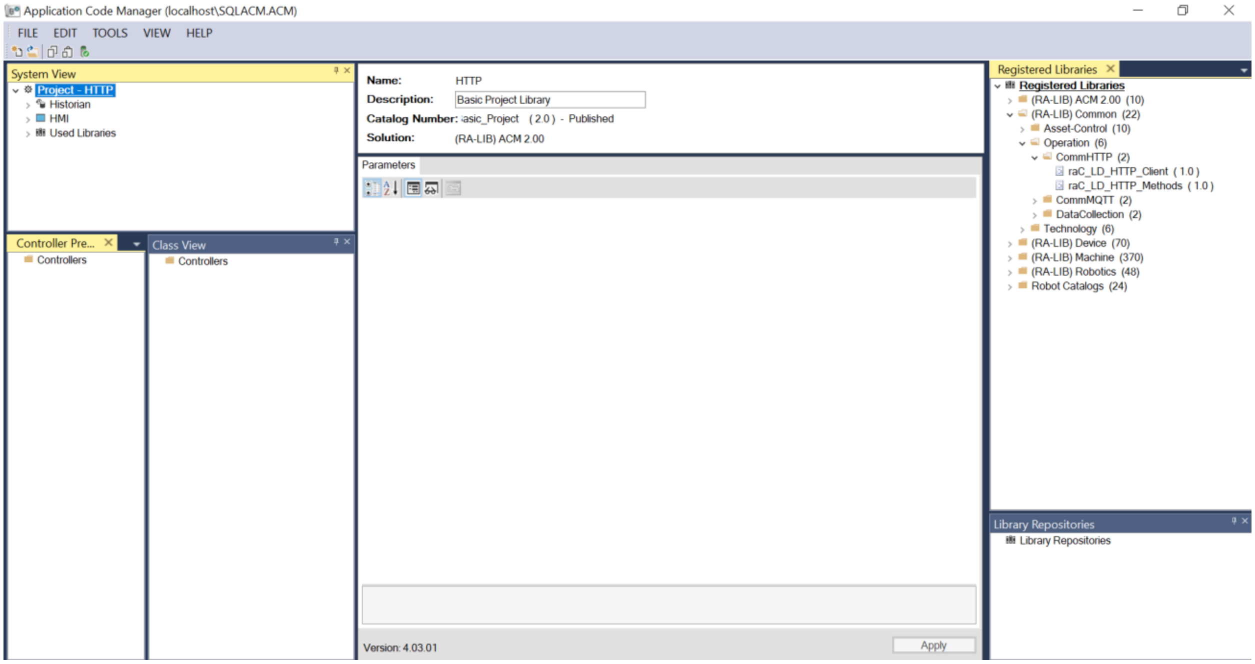
Solution: (RA-LIB) ACM 2.00

Click on finish



Name: HTTP
Description: Basic Project Library
Catalog Number: ꞉asic_Project (2.0) - Published
Solution: (RA-LIB) ACM 2.00

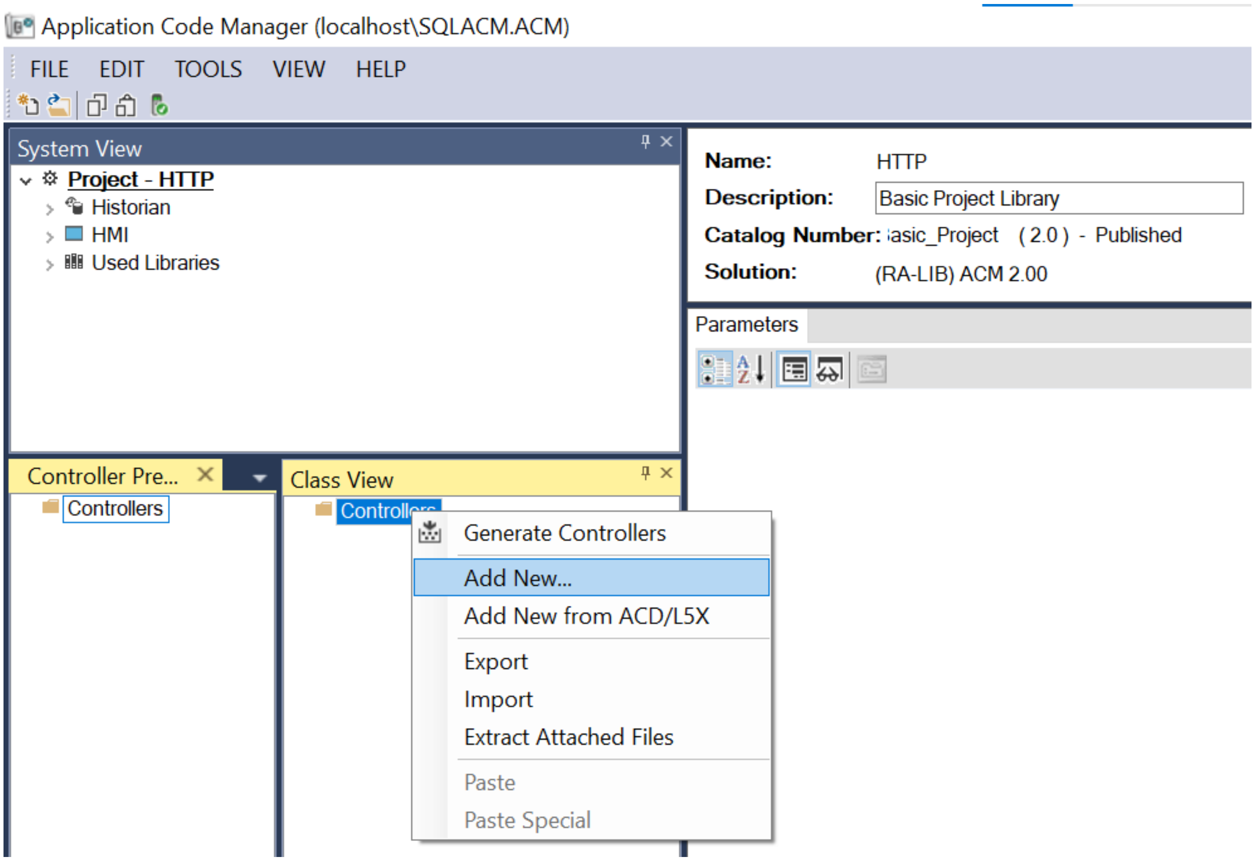
Now you have this blank work space



Name: HTTP
Description: Basic Project Library
Catalog Number: ꞉asic_Project (2.0) - Published
Solution: (RA-LIB) ACM 2.00

Version: 4.03.01

Add a new controller


Application Code Manager (localhost\SQLACM.ACM)

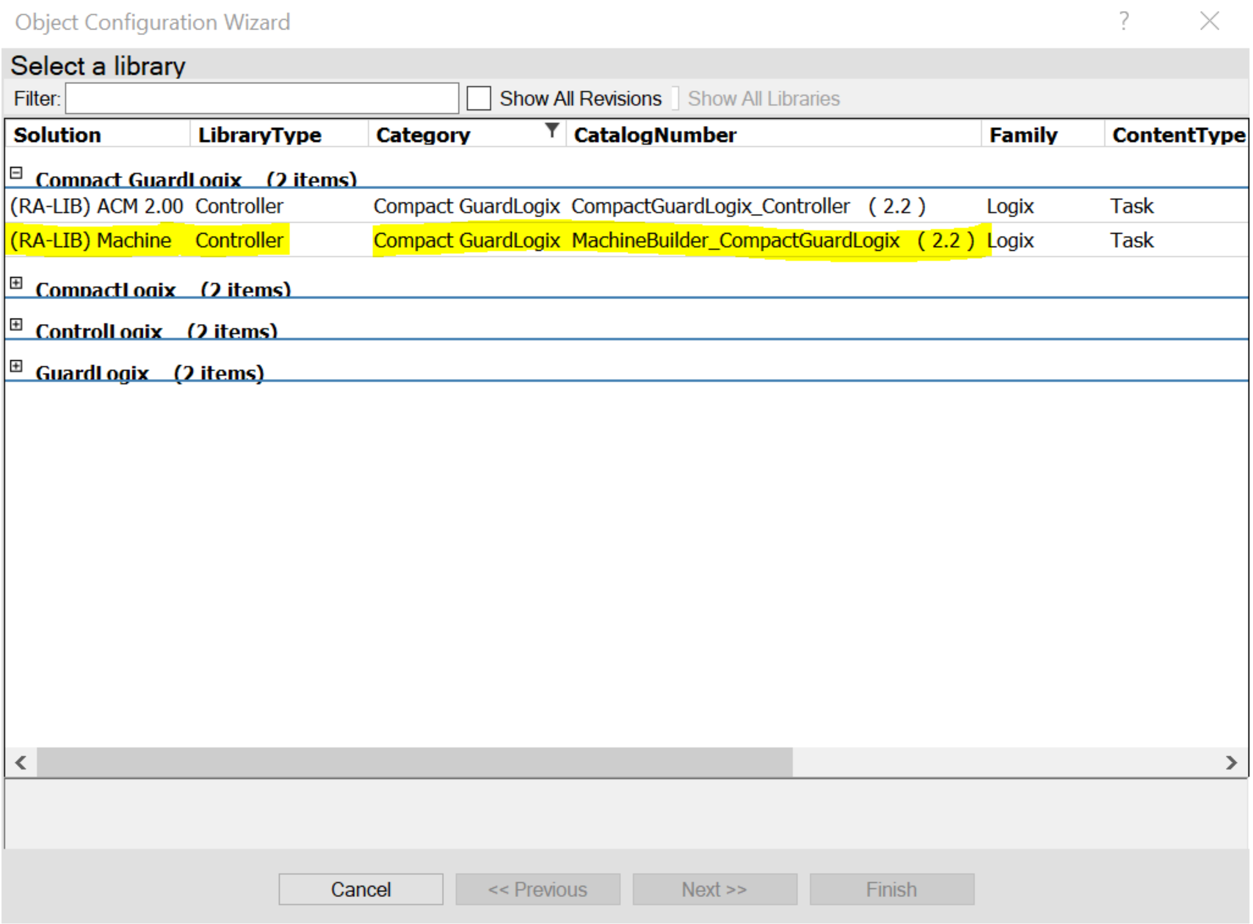
Name: HTTP

Description: Basic Project Library

Catalog Number: ıasic_Project (2.0) - Published

Solution: (RA-LIB) ACM 2.00

Choose your PLC with Machine option (this will create the tasks)

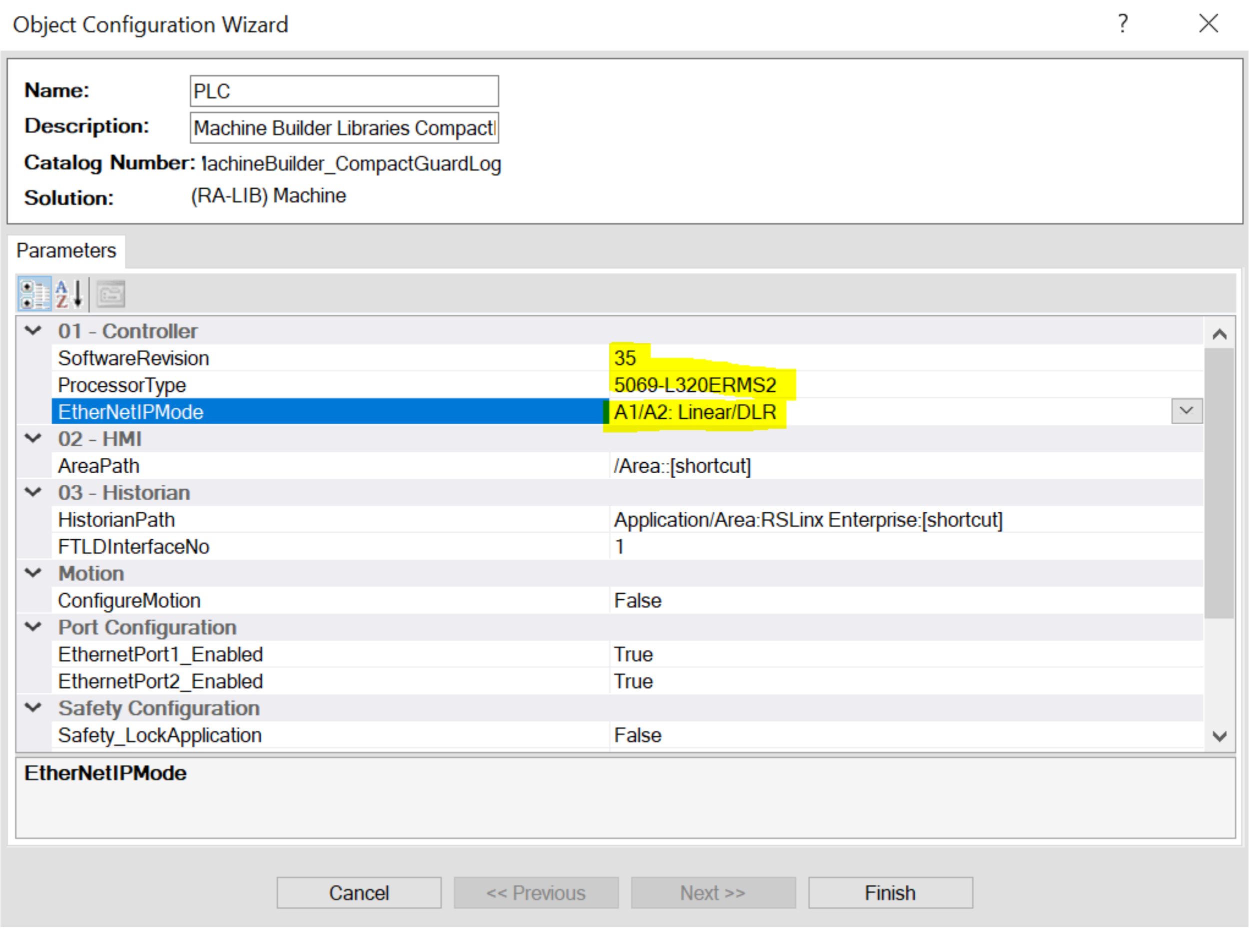


Select a library

Filter: □ Show All Revisions Show All Libraries

Solution	LibraryType	Category	CatalogNumber	Family	ContentType
⊟ Compact GuardLogix (2 items)					
(RA-LIB) ACM 2.00	Controller	Compact GuardLogix	CompactGuardLogix_Controller (2.2)	Logix	Task
(RA-LIB) Machine	Controller	Compact GuardLogix	MachineBuilder_CompactGuardLogix (2.2)	Logix	Task
⊞ CompactLogix (2 items)					
⊞ ControlLogix (2 items)					
⊞ GuardLogix (2 items)					

Give a name, software revision, controller type and Ethernet Port configuration. Click on finish



Name: PLC

Description: Machine Builder Libraries Compactl

Catalog Number: 1achineBuilder_CompactGuardLog

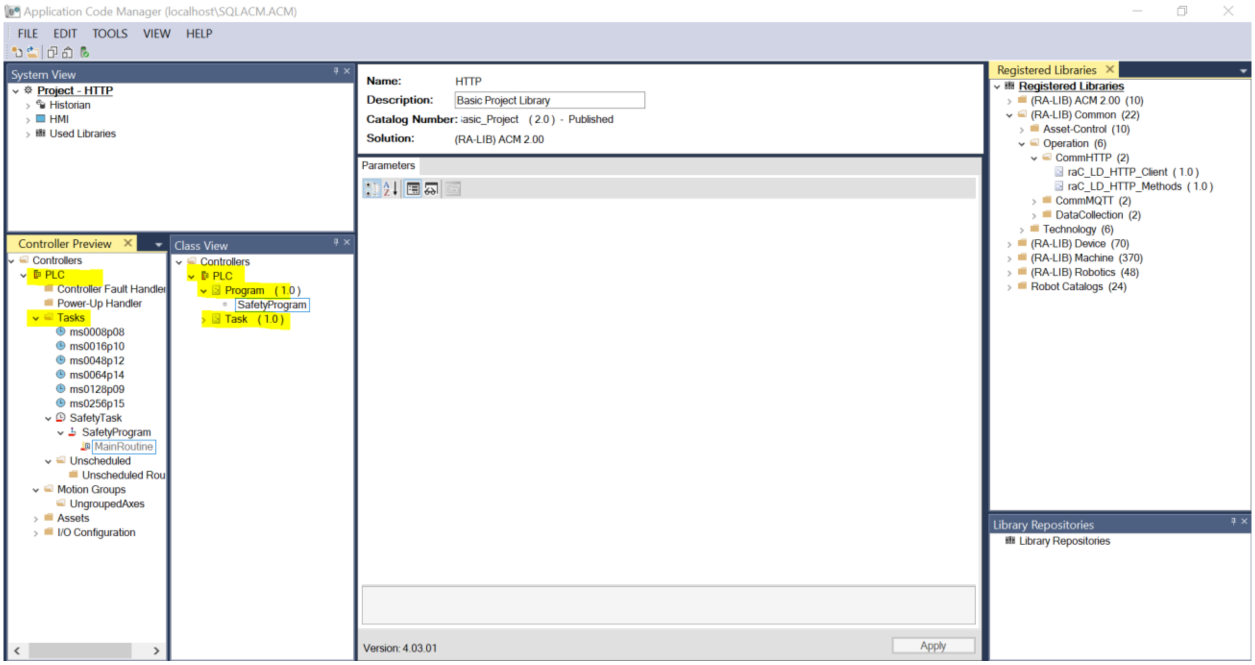
Solution: (RA-LIB) Machine

Parameters

01 - Controller	
SoftwareRevision	35
ProcessorType	5069-L320ERMS2
EtherNetIPMode	A1/A2: Linear/DLR
02 - HMI	
AreaPath	/Area::[shortcut]
03 - Historian	
HistorianPath	Application/Area:RSLinx Enterprise:[shortcut]
FTLDInterfaceNo	1
Motion	
ConfigureMotion	False
Port Configuration	
EthernetPort1_Enabled	True
EthernetPort2_Enabled	True
Safety Configuration	
Safety_LockApplication	False

EtherNetIPMode

The new controller is created, with it's tasks



System View

- Project - HTTP
  - Historian
  - HMI
  - Used Libraries

Controller Preview

- Controllers
  - PLC
    - Controller Fault Handler
    - Power-Up Handler
    - Tasks
      - ms0008p08
      - ms0016p10
      - ms0048p12
      - ms0064p14
      - ms0128p09
      - ms0256p15
      - SafetyTask
        - SafetyProgram
          - MainRoutine
      - Unscheduled
        - Unscheduled Rou
    - Motion Groups
      - UngroupedAxes
    - Assets
    - I/O Configuration

Class View

- Controllers
  - PLC
    - Program (1.0)
      - SafetyProgram
    - Task (1.0)

Name: HTTP

Description: Basic Project Library

Catalog Number: :asic_Project (2.0) - Published

Solution: (RA-LIB) ACM 2.00

Parameters

Version: 4.03.01

Registered Libraries

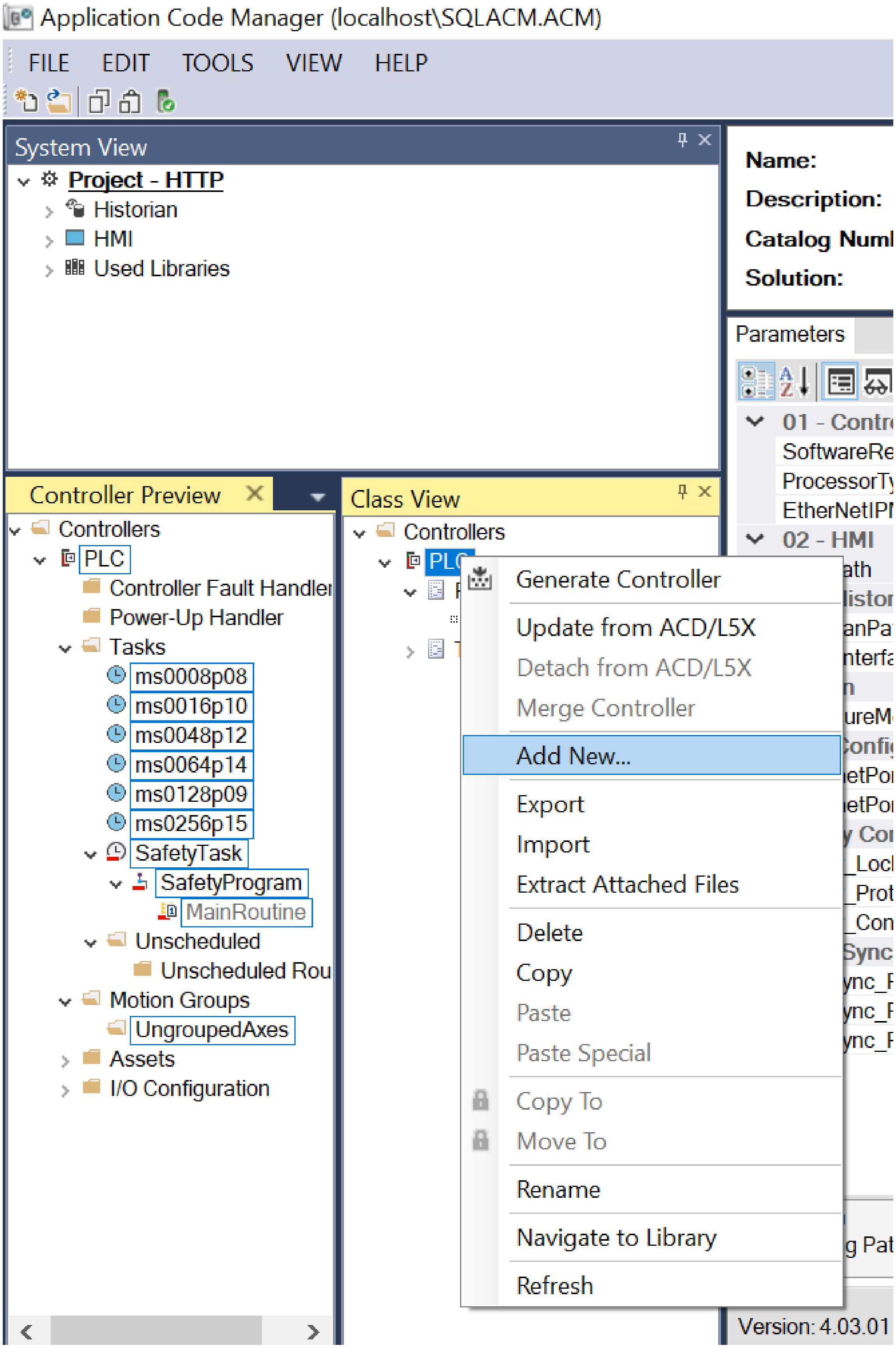
- Registered Libraries
  - (RA-LIB) ACM 2.00 (10)
  - (RA-LIB) Common (22)
    - Asset-Control (10)
    - Operation (6)
      - CommHTTP (2)
        - raC_LD_HTTP_Client (1.0)
        - raC_LD_HTTP_Methods (1.0)
      - CommMQTT (2)
      - DataCollection (2)
    - Technology (6)
  - (RA-LIB) Device (70)
  - (RA-LIB) Machine (370)
  - (RA-LIB) Robotics (48)
  - Robot Catalogs (24)

Library Repositories

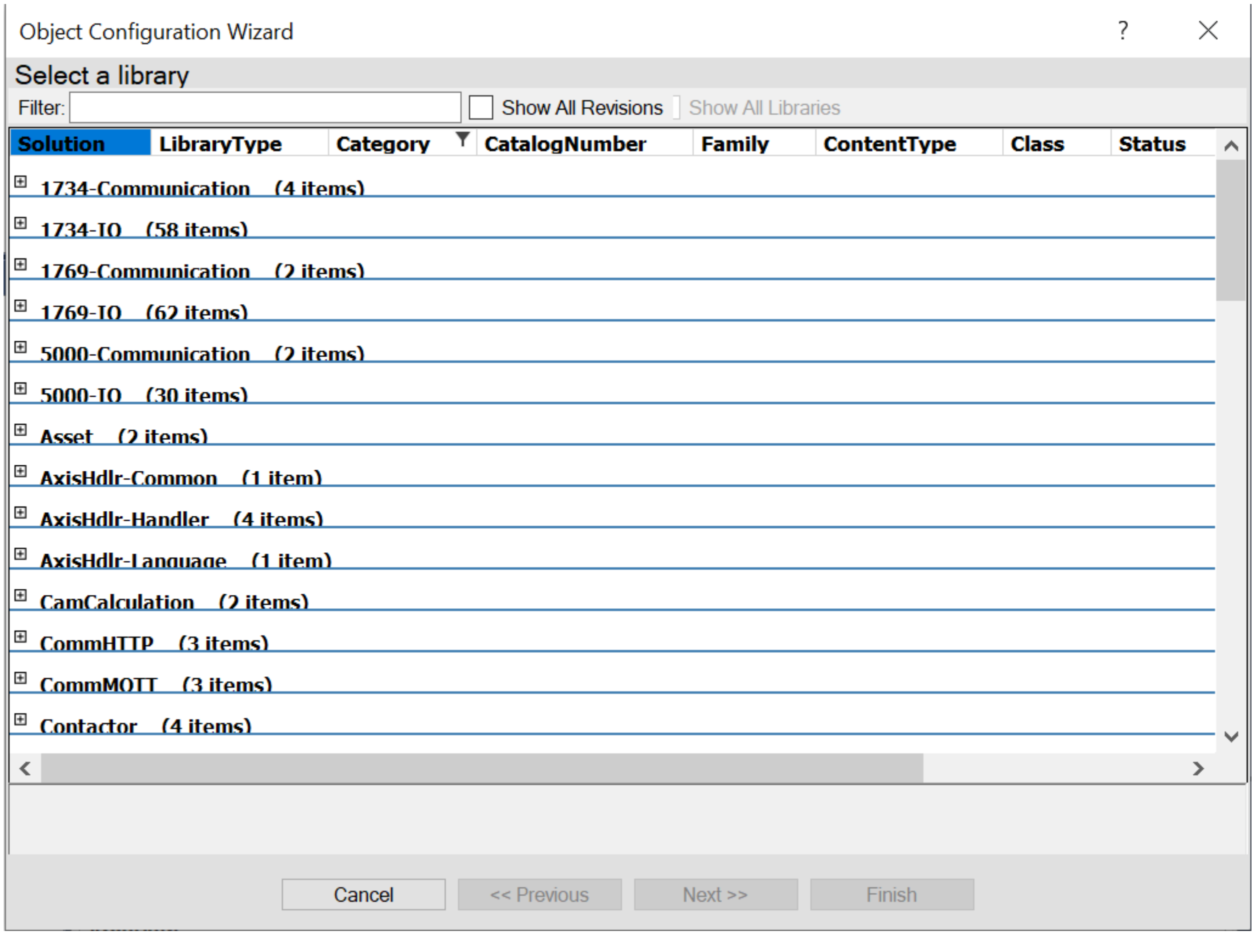
- Library Repositories

Now let’s add the library

Select the controller and right mouse click and select Add


Application Code Manager (localhost\SQLACM.ACM)
FILE
EDIT
TOOLS
VIEW
HELP
System View
Project - HTTP
Historian
HMI
Used Libraries
Name:
Description:
Catalog Numl
Solution:
Parameters
01 - Contr
SoftwareRe
ProcessorTy
EtherNetIPI
02 - HMI
Controller Preview
Controllers
PLC
Controller Fault Handler
Power-Up Handler
Tasks
ms0008p08
ms0016p10
ms0048p12
ms0064p14
ms0128p09
ms0256p15
SafetyTask
SafetyProgram
MainRoutine
Unscheduled
Unscheduled Rou
Motion Groups
UngroupedAxes
Assets
I/O Configuration
Class View
Controllers
PLC
Generate Controller
Update from ACD/L5X
Detach from ACD/L5X
Merge Controller
Add New...
Export
Import
Extract Attached Files
Delete
Copy
Paste
Paste Special
Copy To
Move To
Rename
Navigate to Library
Refresh
Version: 4.03.01

And select a library



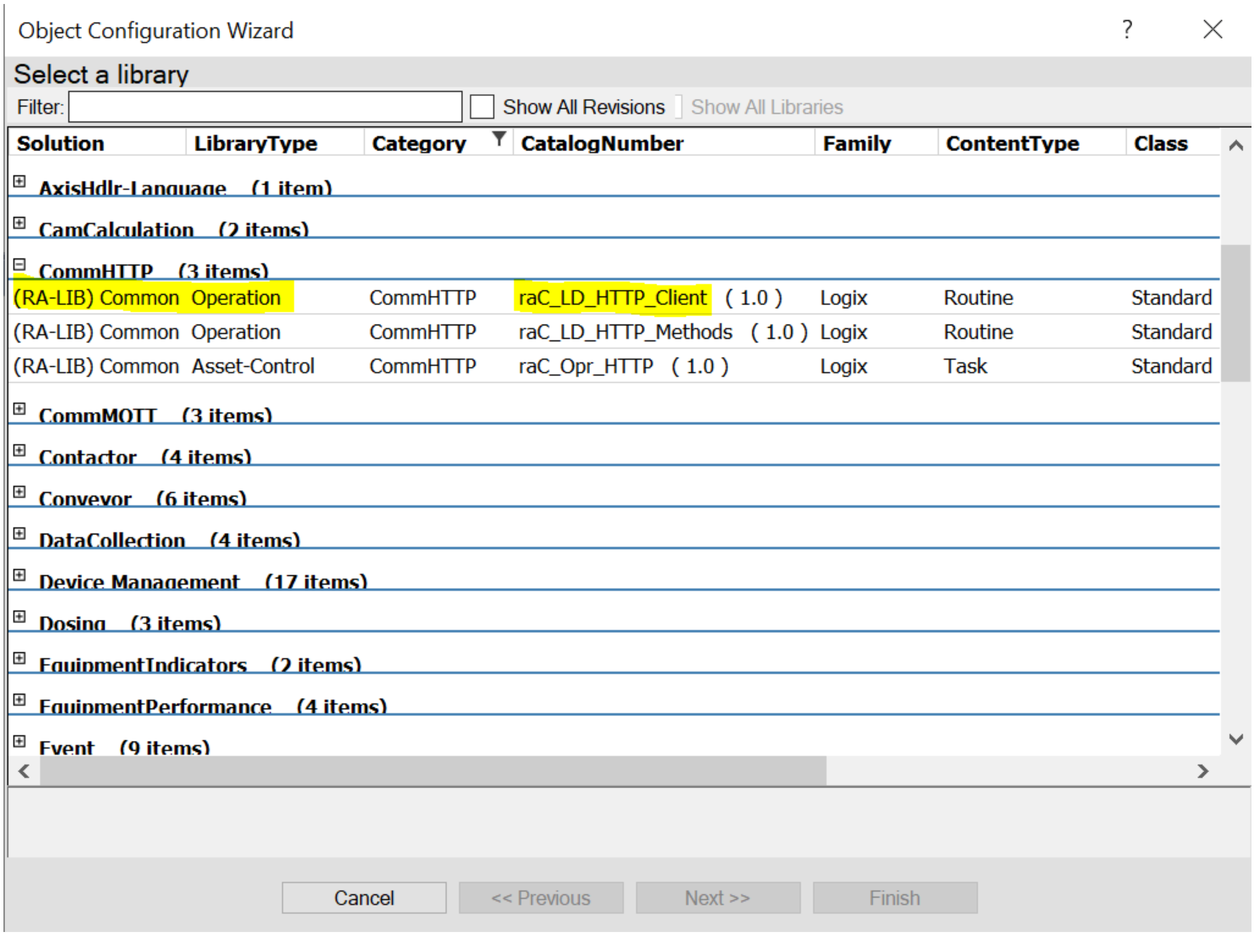
Object Configuration Wizard

Select a library

Filter: □ Show All Revisions Show All Libraries

Solution	LibraryType	Category	CatalogNumber	Family	ContentType	Class	Status
⊞ 1734-Communication (4 items)							
⊞ 1734-IO (58 items)							
⊞ 1769-Communication (2 items)							
⊞ 1769-IO (62 items)							
⊞ 5000-Communication (2 items)							
⊞ 5000-IO (30 items)							
⊞ Asset (2 items)							
⊞ AxisHdlr-Common (1 item)							
⊞ AxisHdlr-Handler (4 items)							
⊞ AxisHdlr-Language (1 item)							
⊞ CamCalculation (2 items)							
⊞ CommHTTP (3 items)							
⊞ CommMQTT (3 items)							
⊞ Contactor (4 items)							

Select the client library

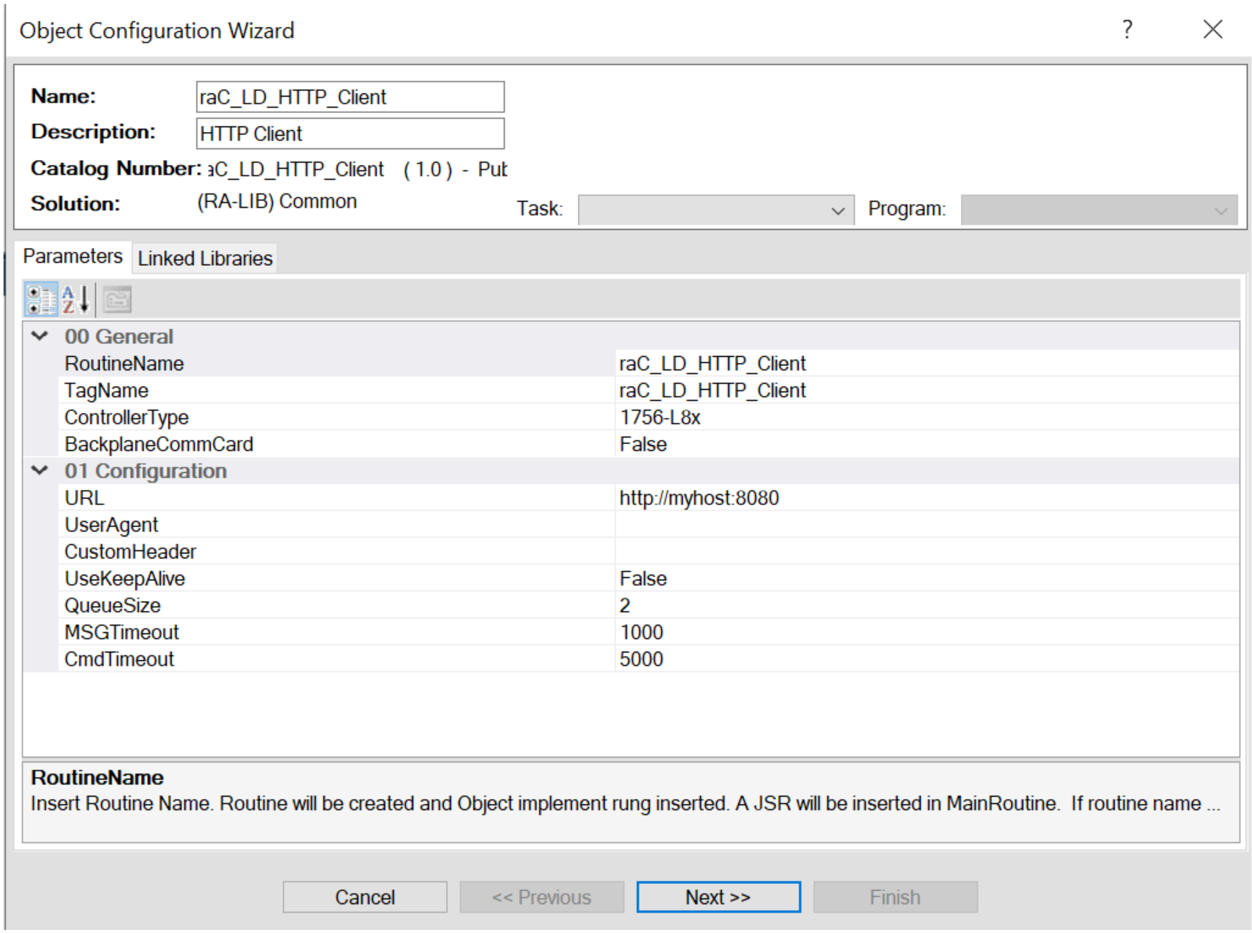


Select a library

Filter: □ Show All Revisions Show All Libraries

Solution	LibraryType	Category	CatalogNumber	Family	ContentType	Class
⊞ AxisHdlr-Language (1 item)						
⊞ CamCalculation (2 items)						
⊟ CommHTTP (3 items)						
(RA-LIB) Common	Operation	CommHTTP	raC_LD_HTTP_Client (1.0)	Logix	Routine	Standard
(RA-LIB) Common	Operation	CommHTTP	raC_LD_HTTP_Methods (1.0)	Logix	Routine	Standard
(RA-LIB) Common	Asset-Control	CommHTTP	raC_Opr_HTTP (1.0)	Logix	Task	Standard
⊞ CommMQTT (3 items)						
⊞ Contactor (4 items)						
⊞ Conveyor (6 items)						
⊞ DataCollection (4 items)						
⊞ Device Management (17 items)						
⊞ Dosing (3 items)						
⊞ EquipmentIndicators (2 items)						
⊞ EquipmentPerformance (4 items)						
⊞ Event (9 items)						

Click on next



Name: raC_LD_HTTP_Client

Description: HTTP Client

Catalog Number: aC_LD_HTTP_Client (1.0) - Put

Solution: (RA-LIB) Common

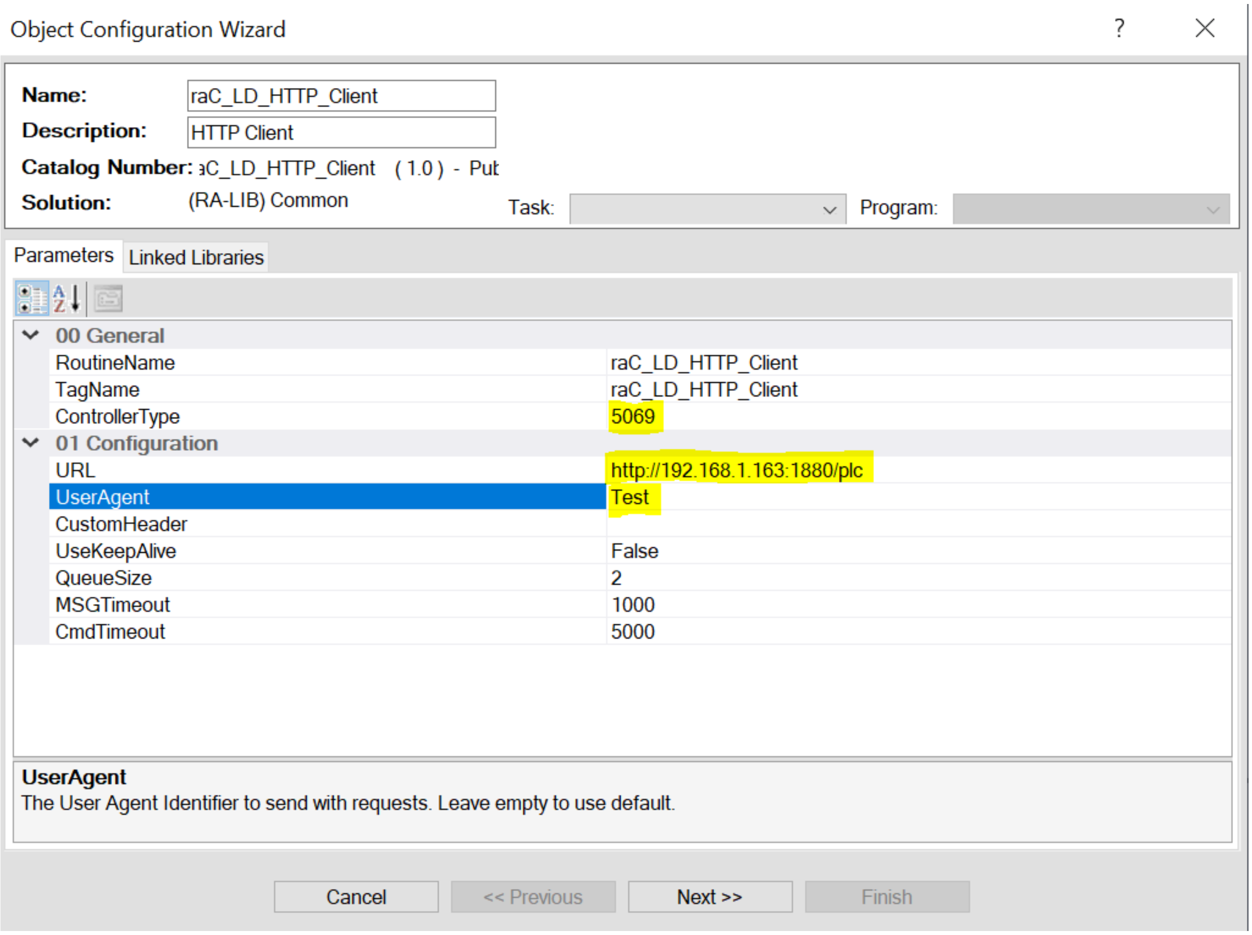
Task:

Program:

00 General	
RoutineName	raC_LD_HTTP_Client
TagName	raC_LD_HTTP_Client
ControllerType	1756-L8x
BackplaneCommCard	False
01 Configuration	
URL	http://myhost:8080
UserAgent	
CustomHeader	
UseKeepAlive	False
QueueSize	2
MSGTimeout	1000
CmdTimeout	5000

RoutineName
Insert Routine Name. Routine will be created and Object implement rung inserted. A JSR will be inserted in MainRoutine. If routine name ...

Enter the parameters like controller type, url (where the API Rest server lies, we will use node-red), etc

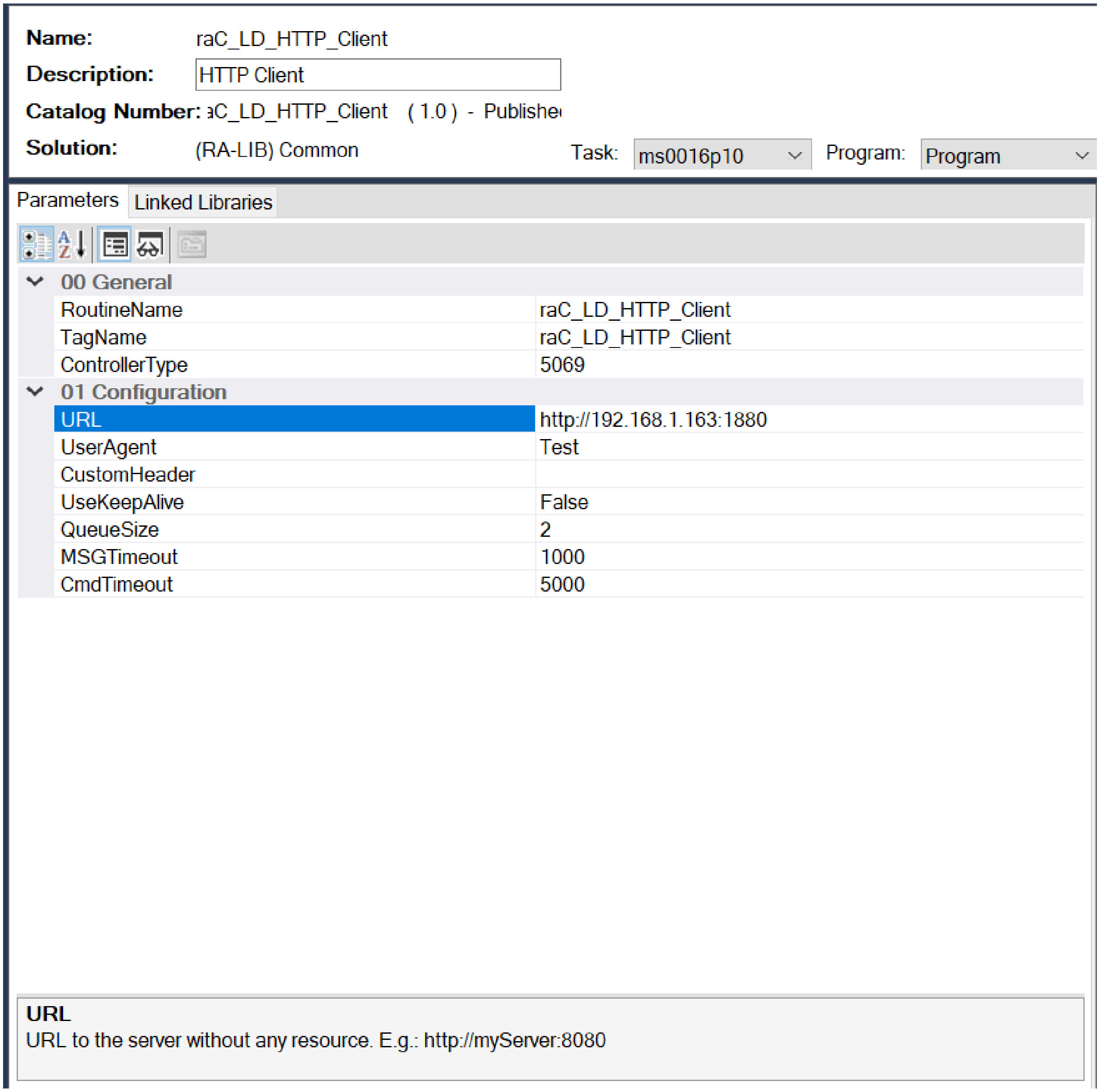


Name: raC_LD_HTTP_Client
Description: HTTP Client
Catalog Number: ɜC_LD_HTTP_Client (1.0) - Put
Solution: (RA-LIB) Common
Task:
Program:

00 General	
RoutineName	raC_LD_HTTP_Client
TagName	raC_LD_HTTP_Client
ControllerType	5069
01 Configuration	
URL	http://192.168.1.163:1880/plc
UserAgent	Test
CustomHeader	
UseKeepAlive	False
QueueSize	2
MSGTimeout	1000
CmdTimeout	5000

UserAgent
The User Agent Identifier to send with requests. Leave empty to use default.

Attention, the URL is without any resource. Correct this



Name: raC_LD_HTTP_Client

Description: HTTP Client

Catalog Number: aC_LD_HTTP_Client (1.0) - Publishe

Solution: (RA-LIB) Common

Task: ms0016p10 Program: Program

00 General

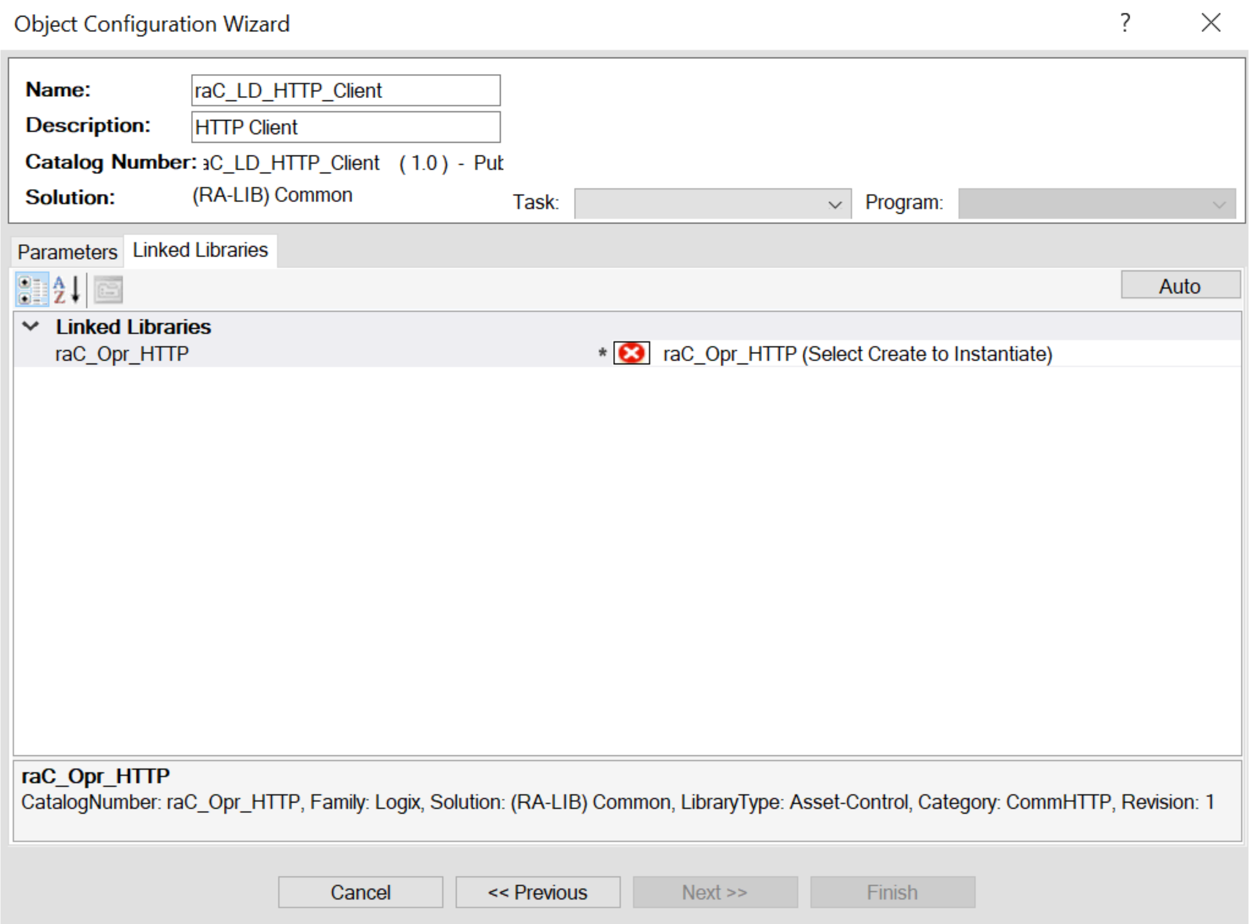
RoutineName	raC_LD_HTTP_Client
TagName	raC_LD_HTTP_Client
ControllerType	5069

01 Configuration

URL	http://192.168.1.163:1880
UserAgent	Test
CustomHeader	
UseKeepAlive	False
QueueSize	2
MSGTimeout	1000
CmdTimeout	5000

URL
URL to the server without any resource. E.g.: http://myServer:8080

Click on Next



Object Configuration Wizard

Name: raC_LD_HTTP_Client

Description: HTTP Client

Catalog Number: ɜC_LD_HTTP_Client (1.0) - Puł

Solution: (RA-LIB) Common

Task:

Program:

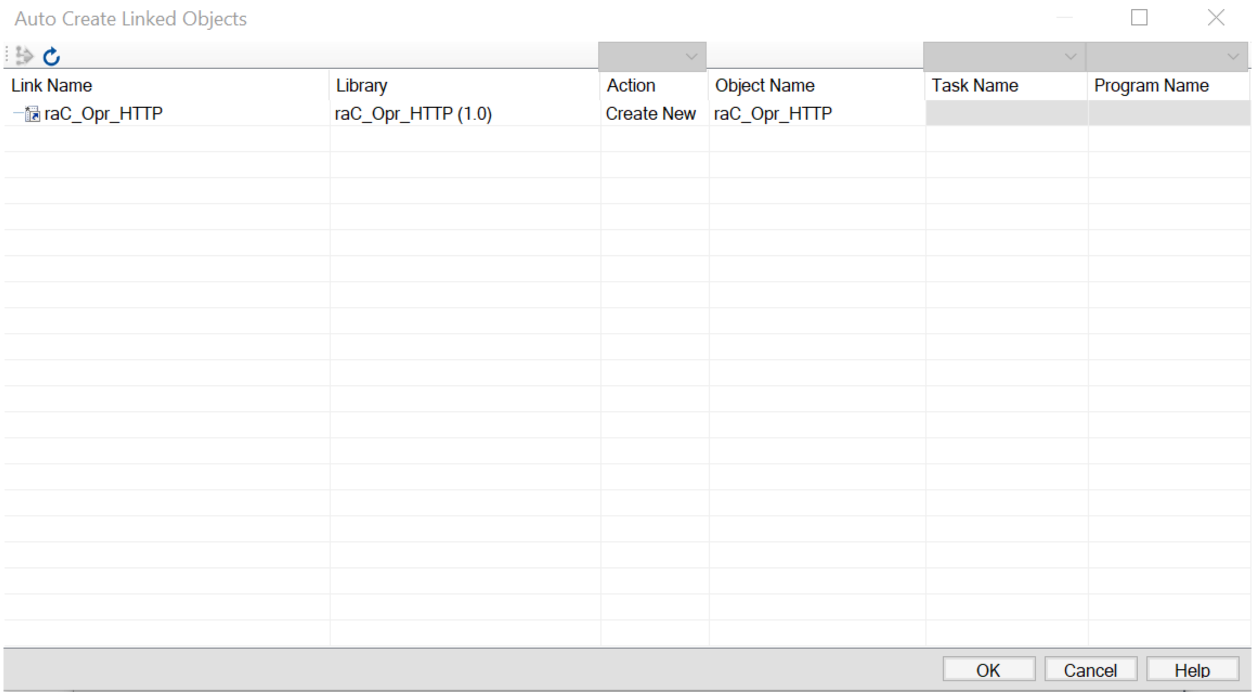
Linked Libraries

raC_Opr_HTTP * ☒ raC_Opr_HTTP (Select Create to Instantiate)

raC_Opr_HTTP

CatalogNumber: raC_Opr_HTTP, Family: Logix, Solution: (RA-LIB) Common, LibraryType: Asset-Control, Category: CommHTTP, Revision: 1

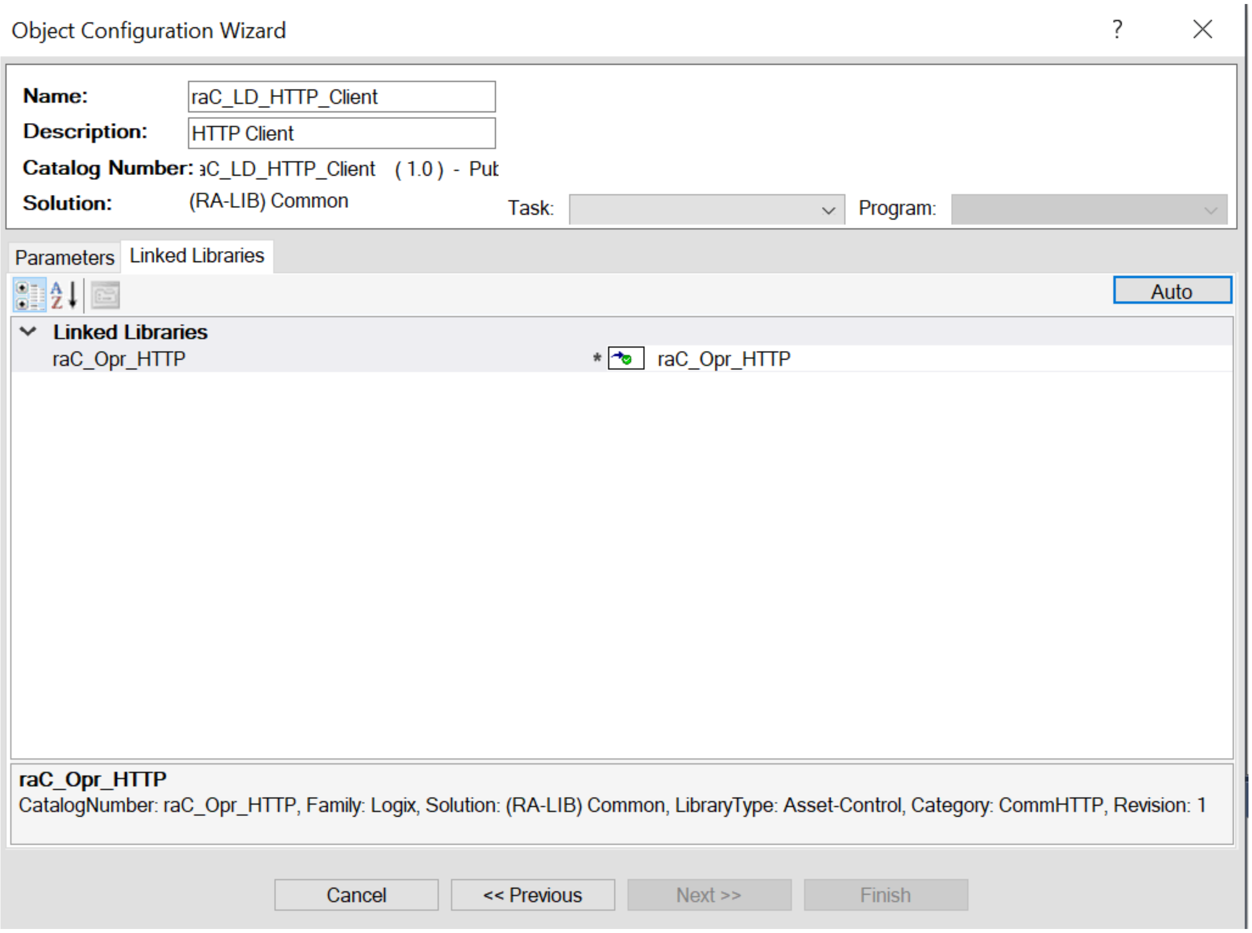
Click on Auto



Auto Create Linked Objects

Link Name	Library	Action	Object Name	Task Name	Program Name
raC_Opr_HTTP	raC_Opr_HTTP (1.0)	Create New	raC_Opr_HTTP		

Click OK to create the instantiation


Object Configuration Wizard

Name: raC_LD_HTTP_Client

Description: HTTP Client

Catalog Number: aC_LD_HTTP_Client (1.0) - Put

Solution: (RA-LIB) Common

Task:

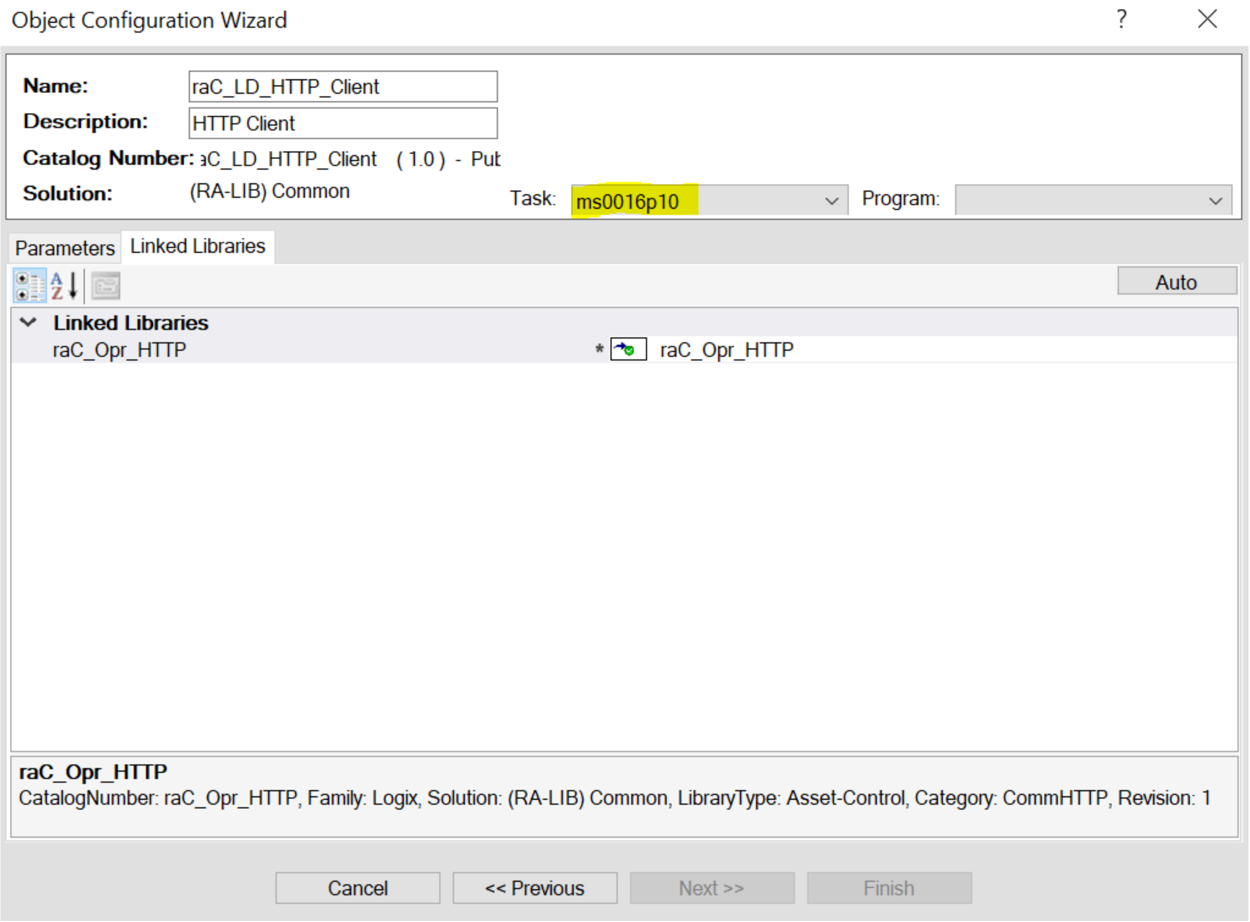
Program:

Linked Libraries	
raC_Opr_HTTP	* raC_Opr_HTTP

raC_Opr_HTTP

CatalogNumber: raC_Opr_HTTP, Family: Logix, Solution: (RA-LIB) Common, LibraryType: Asset-Control, Category: CommHTTP, Revision: 1

Select a task



Object Configuration Wizard

Name: raC_LD_HTTP_Client

Description: HTTP Client

Catalog Number: ɜC_LD_HTTP_Client (1.0) - Put

Solution: (RA-LIB) Common

Task: ms0016p10

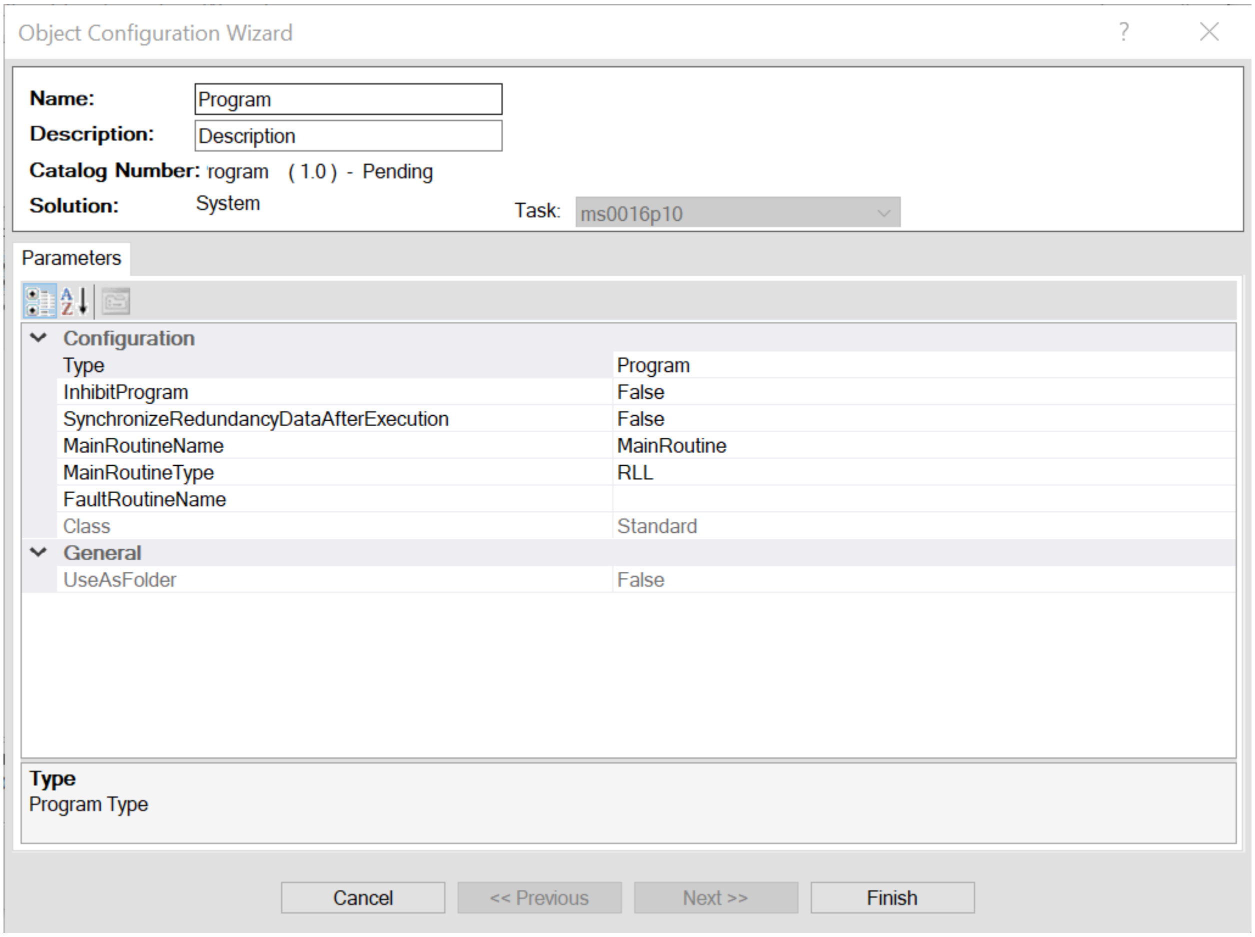
Program:

raC_Opr_HTTP

CatalogNumber: raC_Opr_HTTP, Family: Logix, Solution: (RA-LIB) Common, LibraryType: Asset-Control, Category: CommHTTP, Revision: 1

And create a program

And click on finish



Object Configuration Wizard

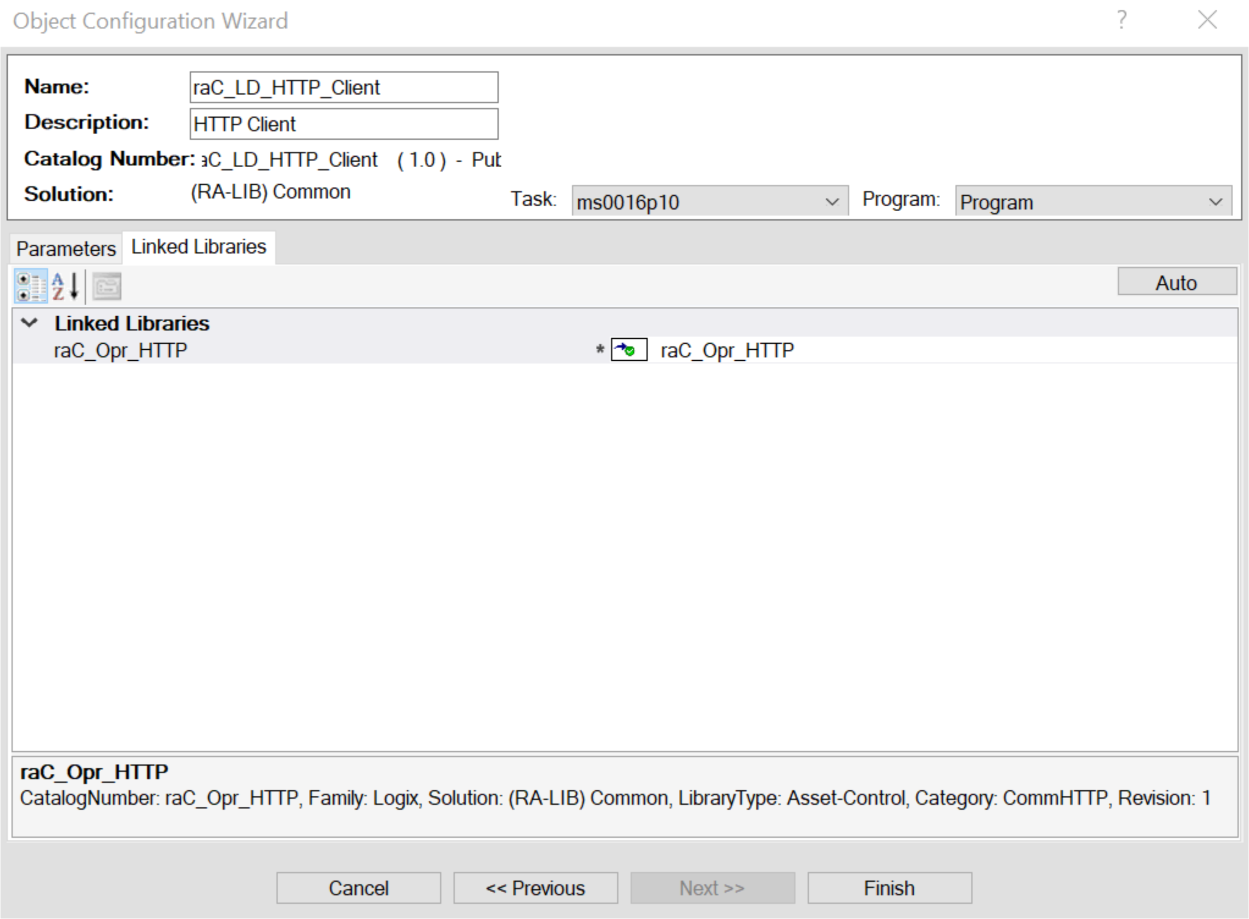
Name: Program
Description: Description
Catalog Number: rogram (1.0) - Pending
Solution: System
Task: ms0016p10

Parameters

Configuration	
Type	Program
InhibitProgram	False
SynchronizeRedundancyDataAfterExecution	False
MainRoutineName	MainRoutine
MainRoutineType	RLL
FaultRoutineName	
Class	Standard
General	
UseAsFolder	False

Type
Program Type

Click on finish



Name: raC_LD_HTTP_Client

Description: HTTP Client

Catalog Number: aC_LD_HTTP_Client (1.0) - Put

Solution: (RA-LIB) Common

Task: ms0016p10

Program: Program

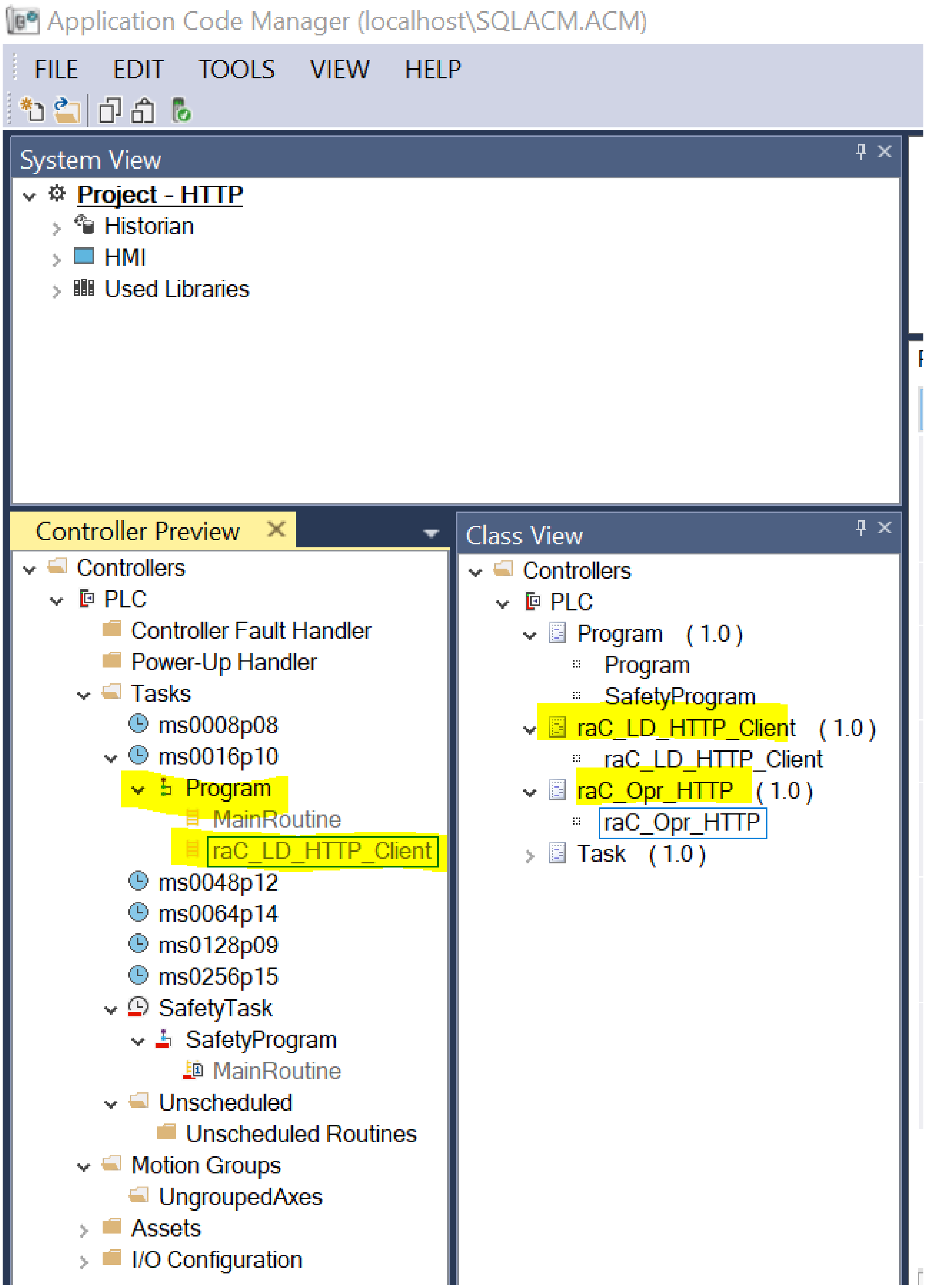
Linked Libraries

raC_Opr_HTTP * raC_Opr_HTTP

raC_Opr_HTTP

CatalogNumber: raC_Opr_HTTP, Family: Logix, Solution: (RA-LIB) Common, LibraryType: Asset-Control, Category: CommHTTP, Revision: 1

Look at the created program



Application Code Manager (localhost\SQLACM.ACM)

System View

- Project - HTTP
  - Historian
  - HMI
  - Used Libraries

Controller Preview

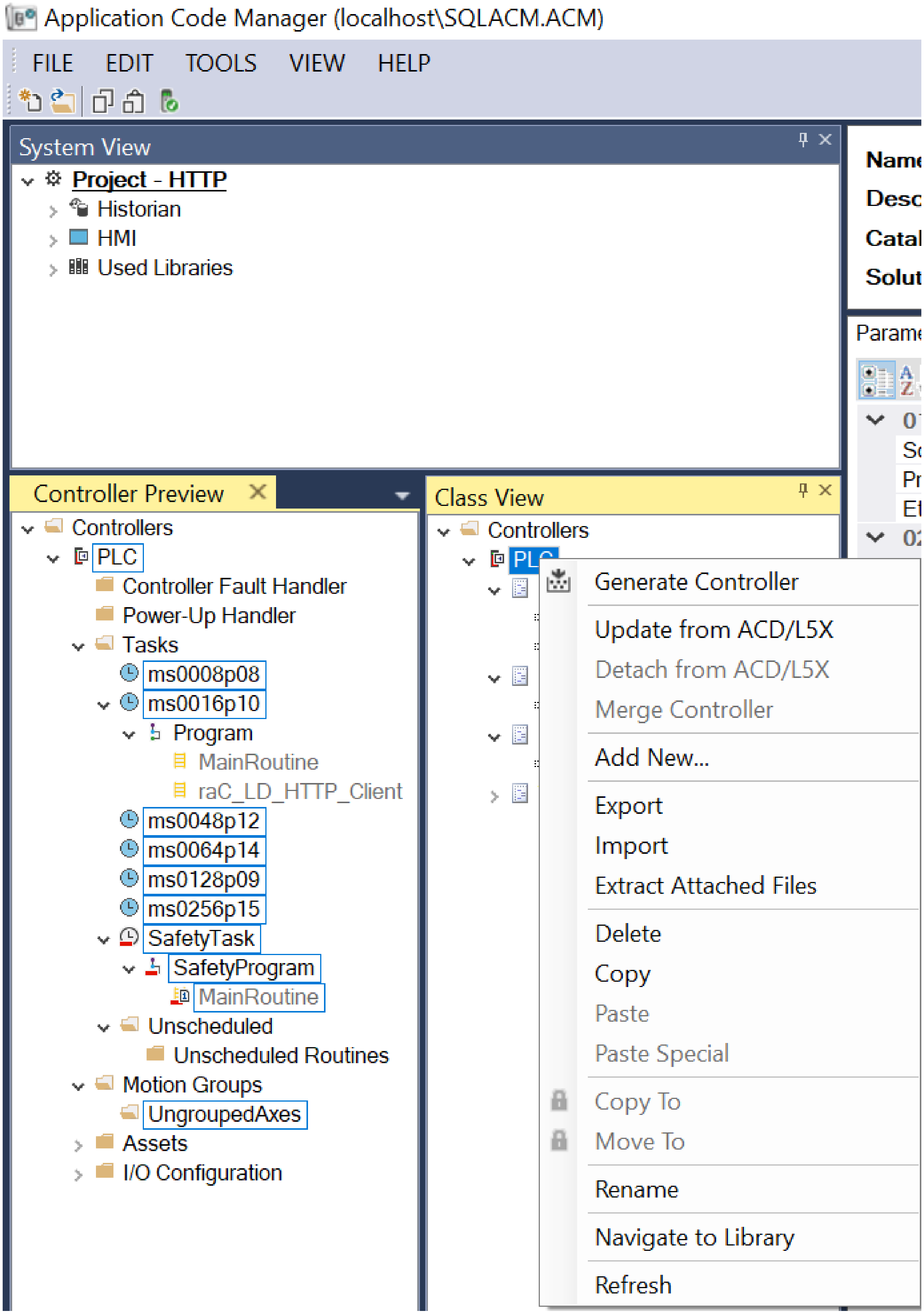
- Controllers
  - PLC
    - Controller Fault Handler
    - Power-Up Handler
    - Tasks
      - ms0008p08
      - ms0016p10
        - Program
          - MainRoutine
          - raC_LD_HTTP_Client
      - ms0048p12
      - ms0064p14
      - ms0128p09
      - ms0256p15
      - SafetyTask
        - SafetyProgram
          - MainRoutine
      - Unscheduled
        - Unscheduled Routines
    - Motion Groups
      - UngroupedAxes
    - Assets
    - I/O Configuration

Class View

- Controllers
  - PLC
    - Program (1.0)
      - Program
      - SafetyProgram
    - raC_LD_HTTP_Client (1.0)
      - raC_LD_HTTP_Client
    - raC_Opr_HTTP (1.0)
      - raC_Opr_HTTP
    - Task (1.0)

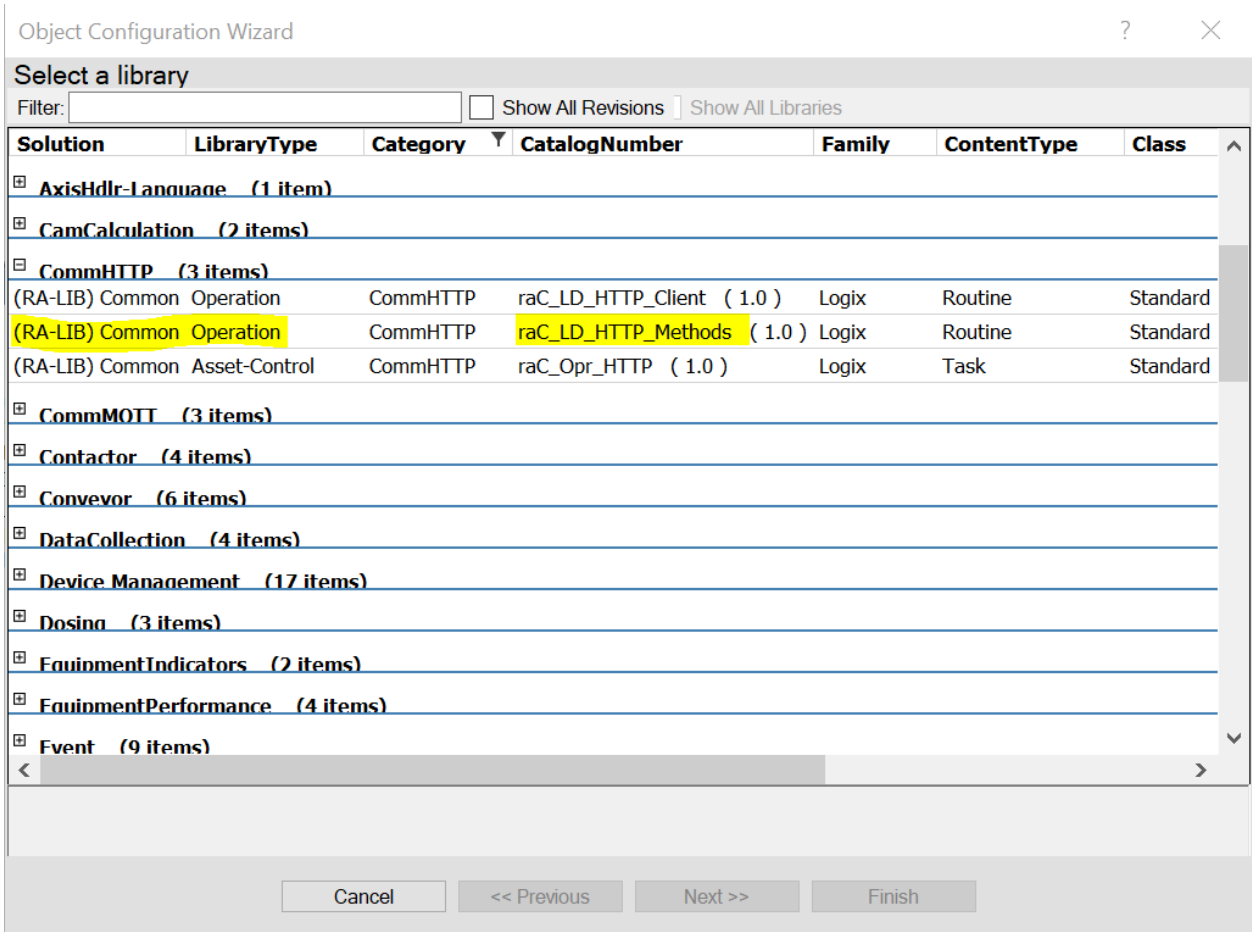
Now let's add the Http Methods on same program

Add New …



Application Code Manager (localhost\SQLACM.ACM)
FILE
EDIT
TOOLS
VIEW
HELP
System View
Project - HTTP
Historian
HMI
Used Libraries
Name
Desc
Catal
Solut
Parame
01
Sc
Pr
Et
02
Controller Preview
Controllers
PLC
Controller Fault Handler
Power-Up Handler
Tasks
ms0008p08
ms0016p10
Program
MainRoutine
raC_LD_HTTP_Client
ms0048p12
ms0064p14
ms0128p09
ms0256p15
SafetyTask
SafetyProgram
MainRoutine
Unscheduled
Unscheduled Routines
Motion Groups
UngroupedAxes
Assets
I/O Configuration
Class View
Controllers
PLC
Generate Controller
Update from ACD/L5X
Detach from ACD/L5X
Merge Controller
Add New...
Export
Import
Extract Attached Files
Delete
Copy
Paste
Paste Special
Copy To
Move To
Rename
Navigate to Library
Refresh

Select and click on Next



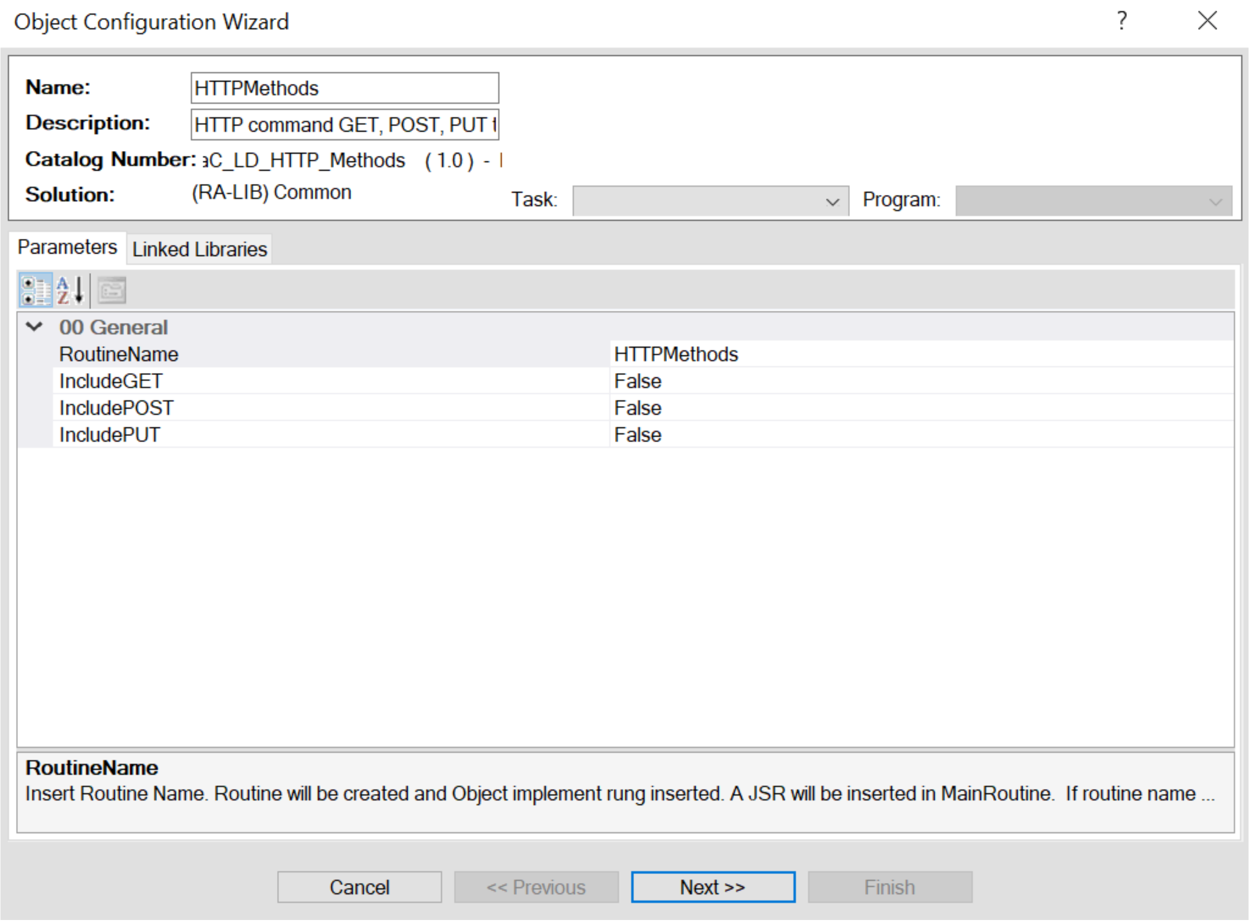
Object Configuration Wizard

Select a library

Filter: □ Show All Revisions Show All Libraries

Solution	LibraryType	Category	CatalogNumber	Family	ContentType	Class
⊞ AxisHdlr-Language (1 item)						
⊞ CamCalculation (2 items)						
⊟ CommHTTP (3 items)						
(RA-LIB) Common	Operation	CommHTTP	raC_LD_HTTP_Client (1.0)	Logix	Routine	Standard
(RA-LIB) Common	Operation	CommHTTP	raC_LD_HTTP_Methods (1.0)	Logix	Routine	Standard
(RA-LIB) Common	Asset-Control	CommHTTP	raC_Opr_HTTP (1.0)	Logix	Task	Standard
⊞ CommMQTT (3 items)						
⊞ Contactor (4 items)						
⊞ Conveyor (6 items)						
⊞ DataCollection (4 items)						
⊞ Device Management (17 items)						
⊞ Dosing (3 items)						
⊞ EquipmentIndicators (2 items)						
⊞ EquipmentPerformance (4 items)						
⊞ Event (9 items)						

Select the same task and program



Object Configuration Wizard

Name: HTTPMethods

Description: HTTP command GET, POST, PUT f

Catalog Number: aC_LD_HTTP_Methods (1.0) - I

Solution: (RA-LIB) Common

Task:

Program:

00 General

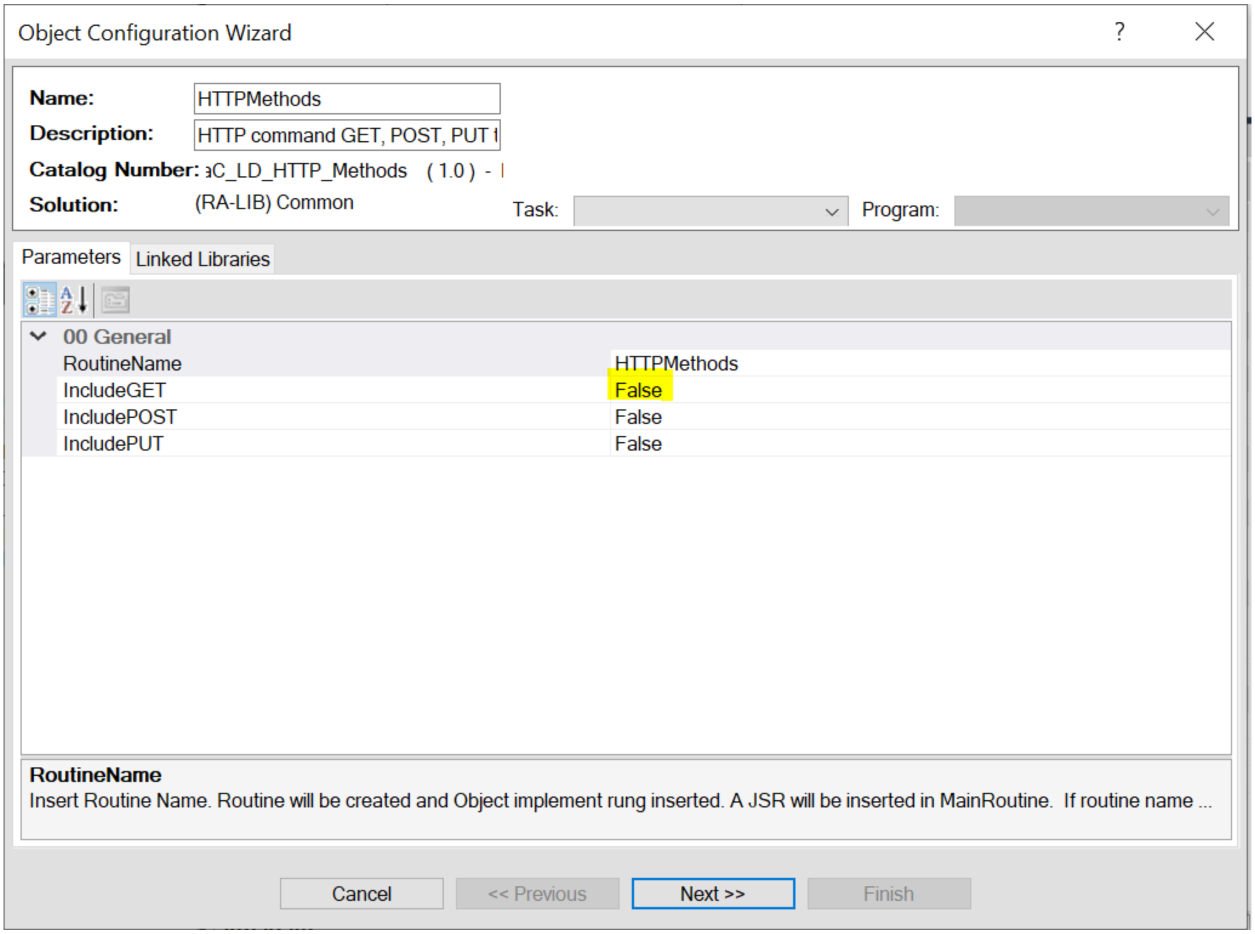
RoutineName	HTTPMethods
IncludeGET	False
IncludePOST	False
IncludePUT	False

RoutineName

Insert Routine Name. Routine will be created and Object implement rung inserted. A JSR will be inserted in MainRoutine. If routine name ...

We will include the GET method

So select True



Name: HTTPMethods

Description: HTTP command GET, POST, PUT t

Catalog Number: ɜC_LD_HTTP_Methods (1.0) - I

Solution: (RA-LIB) Common

Task:

Program:

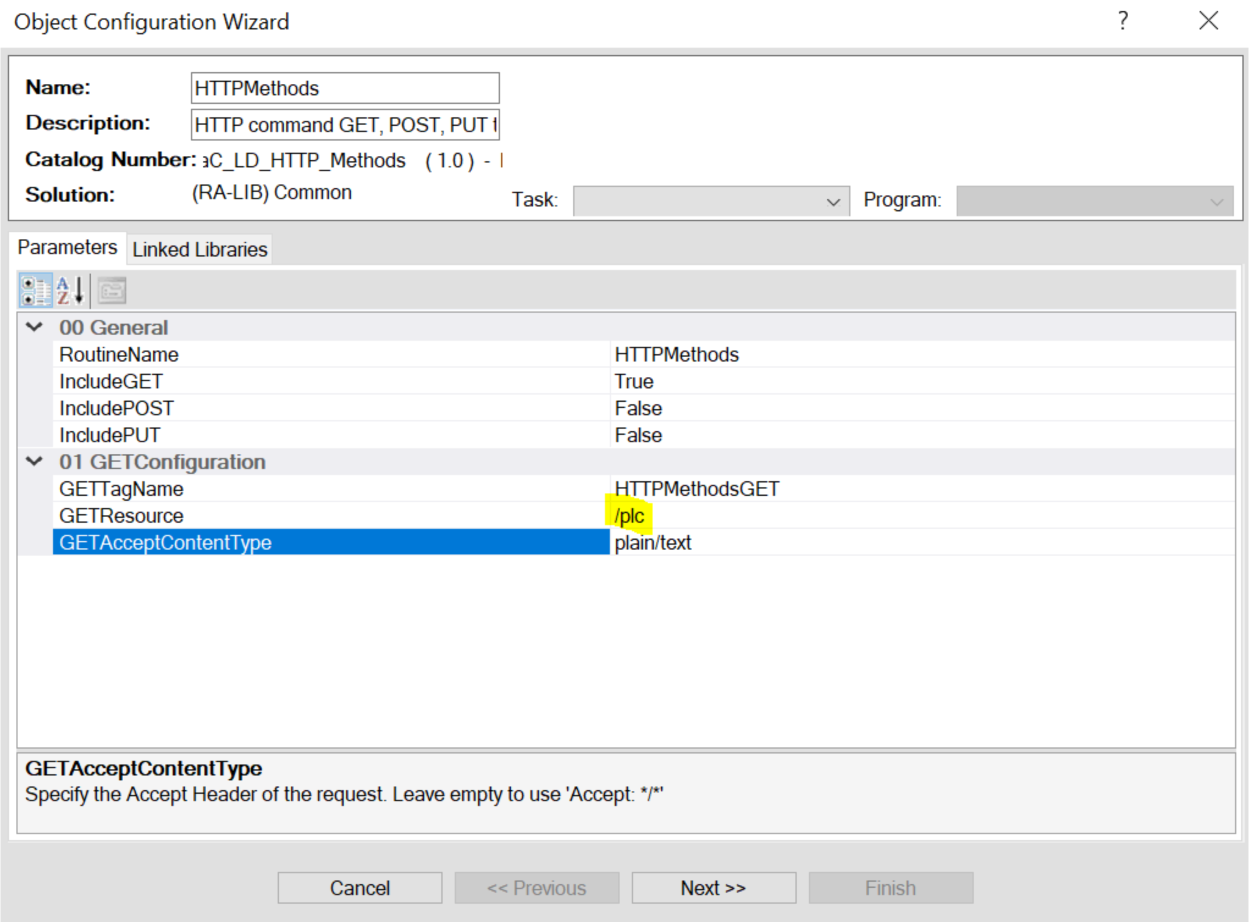
00 General

RoutineName	HTTPMethods
IncludeGET	False
IncludePOST	False
IncludePUT	False

RoutineName

Insert Routine Name. Routine will be created and Object implement rung inserted. A JSR will be inserted in MainRoutine. If routine name ...

And adjust parameters



Object Configuration Wizard

Name: HTTPMethods

Description: HTTP command GET, POST, PUT t

Catalog Number: ɜC_LD_HTTP_Methods (1.0) - I

Solution: (RA-LIB) Common

Task:

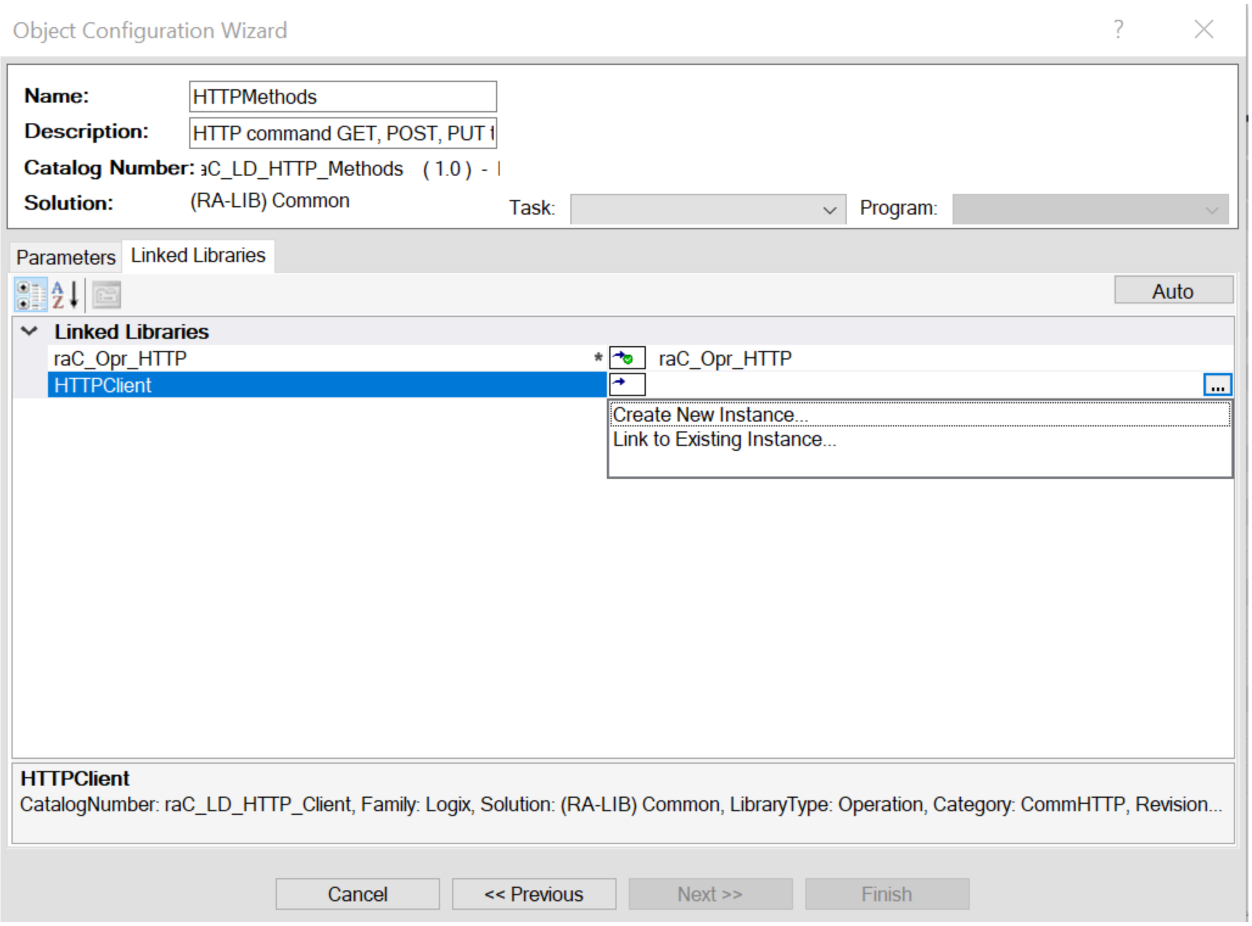
Program:

00 General	
RoutineName	HTTPMethods
IncludeGET	True
IncludePOST	False
IncludePUT	False
01 GETConfiguration	
GETTagName	HTTPMethodsGET
GETResource	/plc
GETAcceptContentType	plain/text

GETAcceptContentType

Specify the Accept Header of the request. Leave empty to use 'Accept: */*'

Linked Libraries



Name: HTTPMethods

Description: HTTP command GET, POST, PUT f

Catalog Number: aC_LD_HTTP_Methods (1.0) - I

Solution: (RA-LIB) Common

Task:

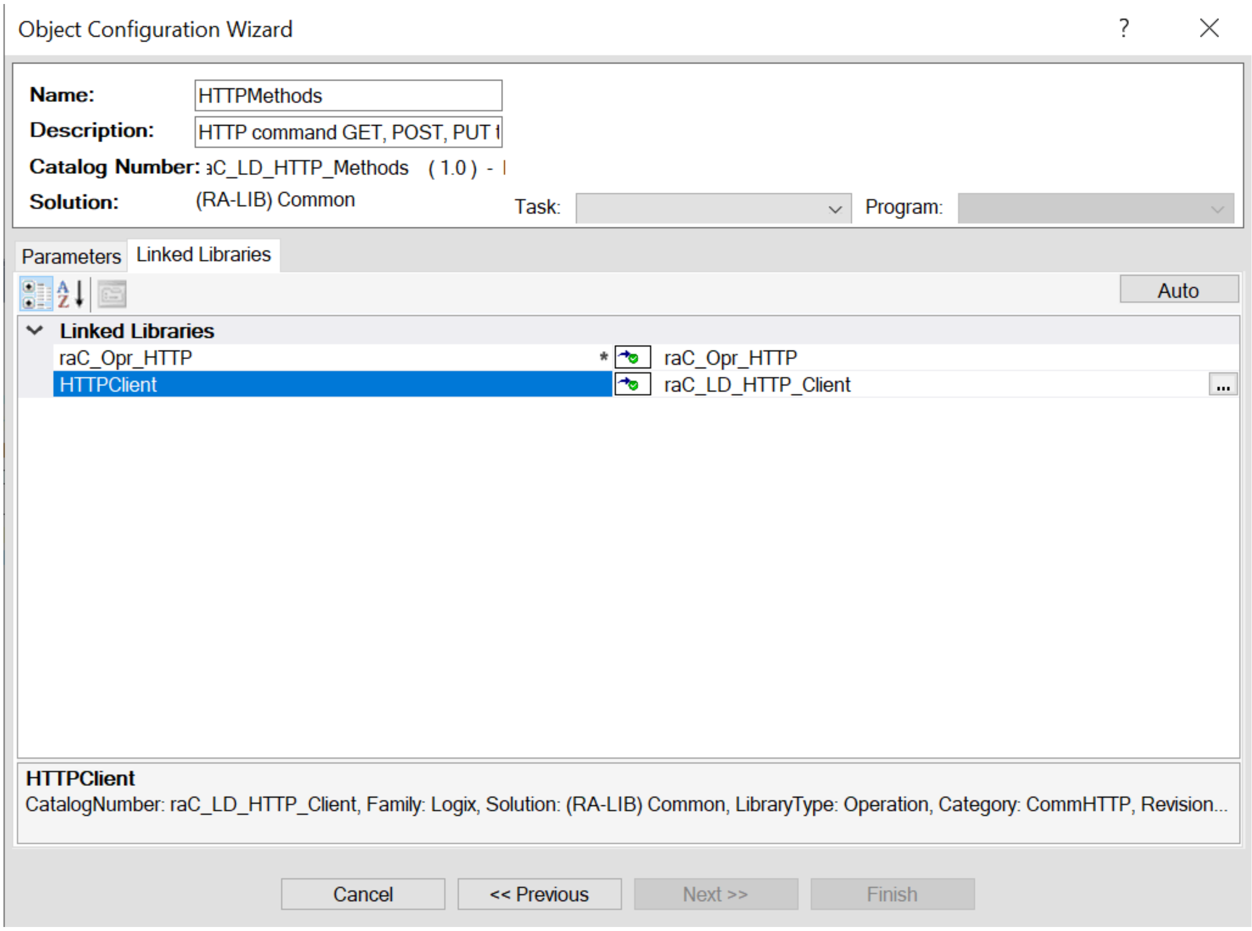
Program:

Linked Libraries

raC_Opr_HTTP *	raC_Opr_HTTP
HTTPClient	...

Create New Instance...
Link to Existing Instance...

HTTPClient
CatalogNumber: raC_LD_HTTP_Client, Family: Logix, Solution: (RA-LIB) Common, LibraryType: Operation, Category: CommHTTP, Revision...



Name: HTTPMethods

Description: HTTP command GET, POST, PUT t

Catalog Number: aC_LD_HTTP_Methods (1.0) - I

Solution: (RA-LIB) Common

Task:

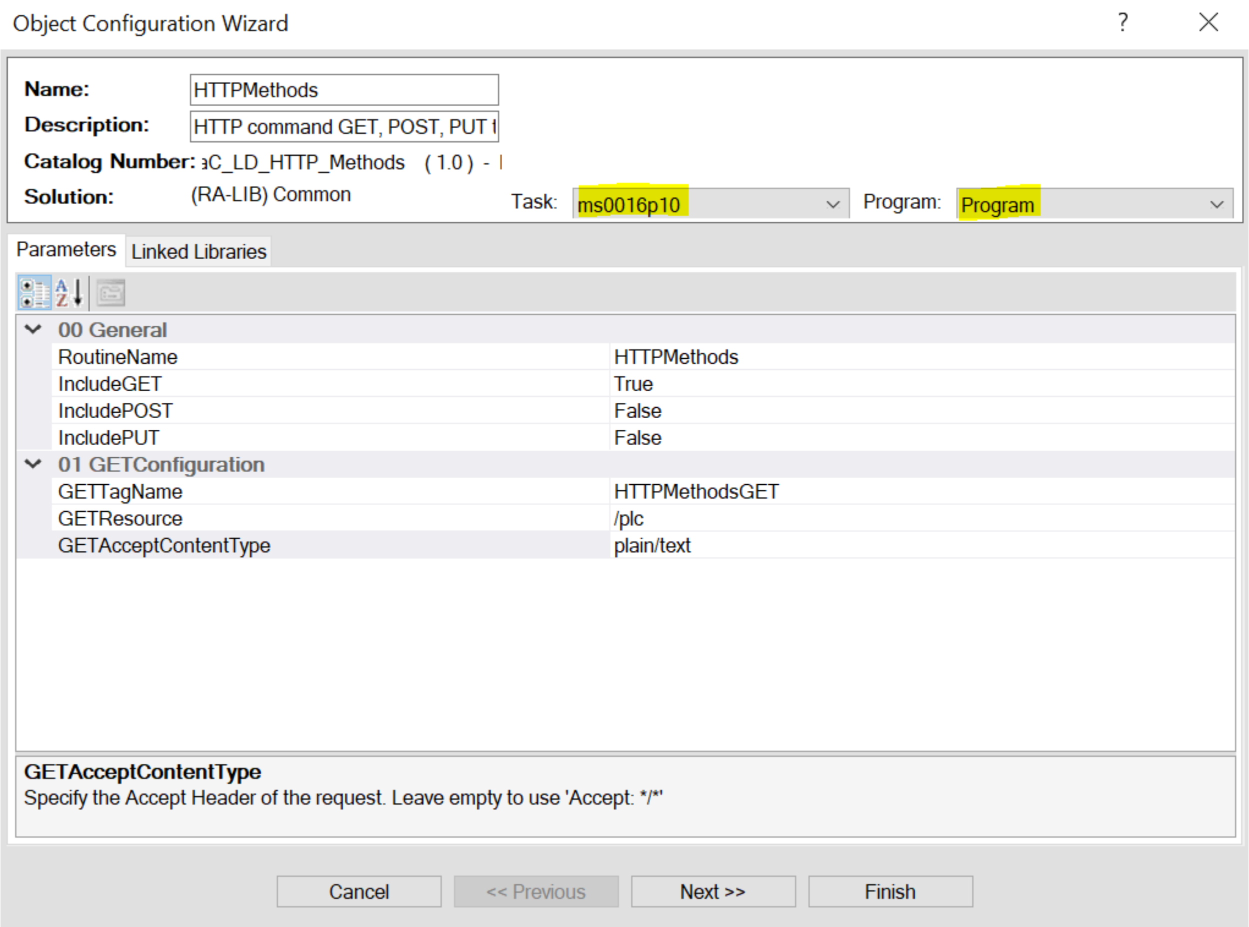
Program:

Linked Libraries

raC_Opr_HTTP *	raC_Opr_HTTP
HTTPClient	raC_LD_HTTP_Client

HTTPClient

CatalogNumber: raC_LD_HTTP_Client, Family: Logix, Solution: (RA-LIB) Common, LibraryType: Operation, Category: CommHTTP, Revision...



Object Configuration Wizard

Name: HTTPMethods

Description: HTTP command GET, POST, PUT f

Catalog Number: aC_LD_HTTP_Methods (1.0) - I

Solution: (RA-LIB) Common

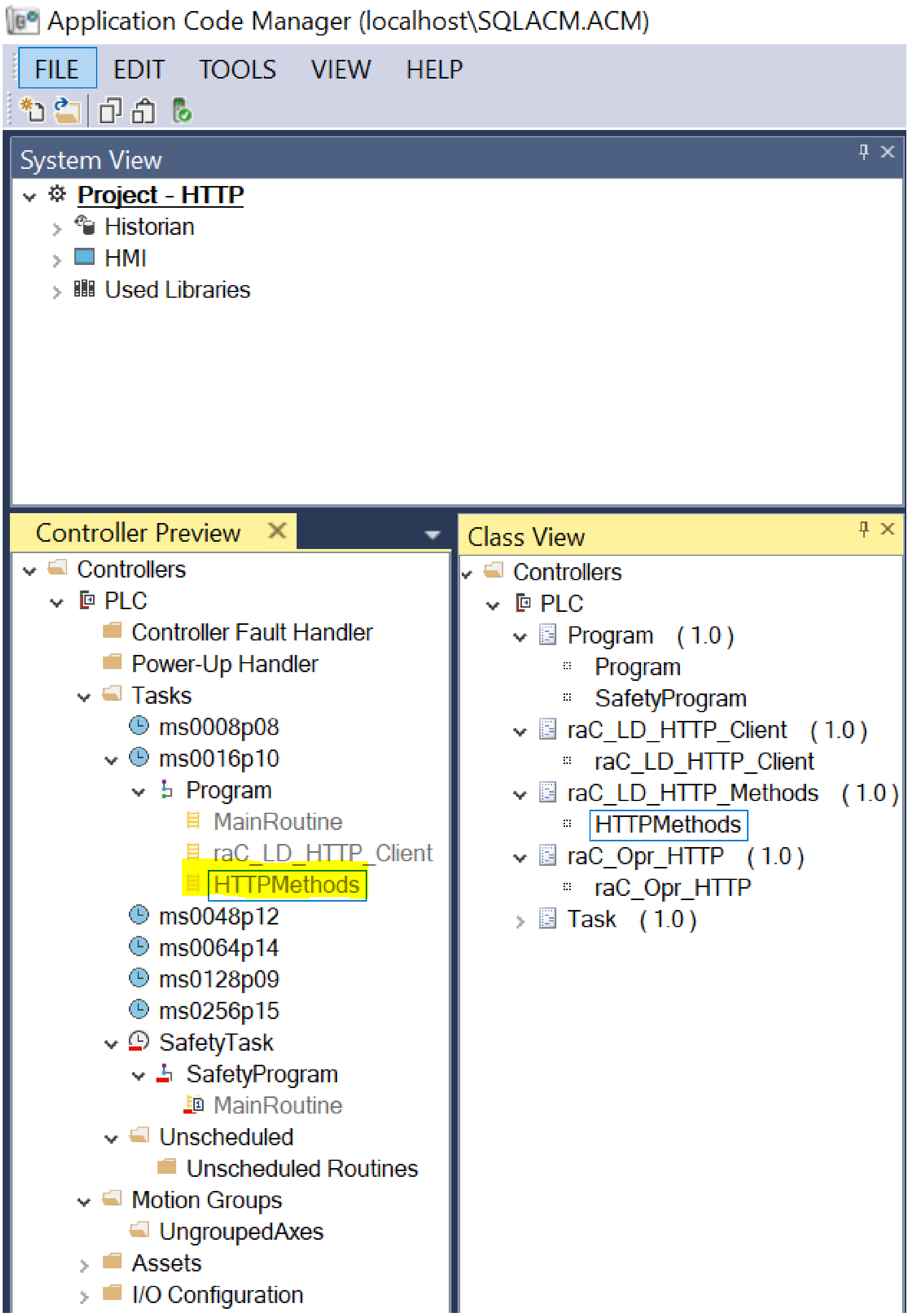
Task: ms0016p10

Program: Program

00 General	
RoutineName	HTTPMethods
IncludeGET	True
IncludePOST	False
IncludePUT	False
01 GETConfiguration	
GETTagName	HTTPMethodsGET
GETResource	/plc
GETAcceptContentType	plain/text

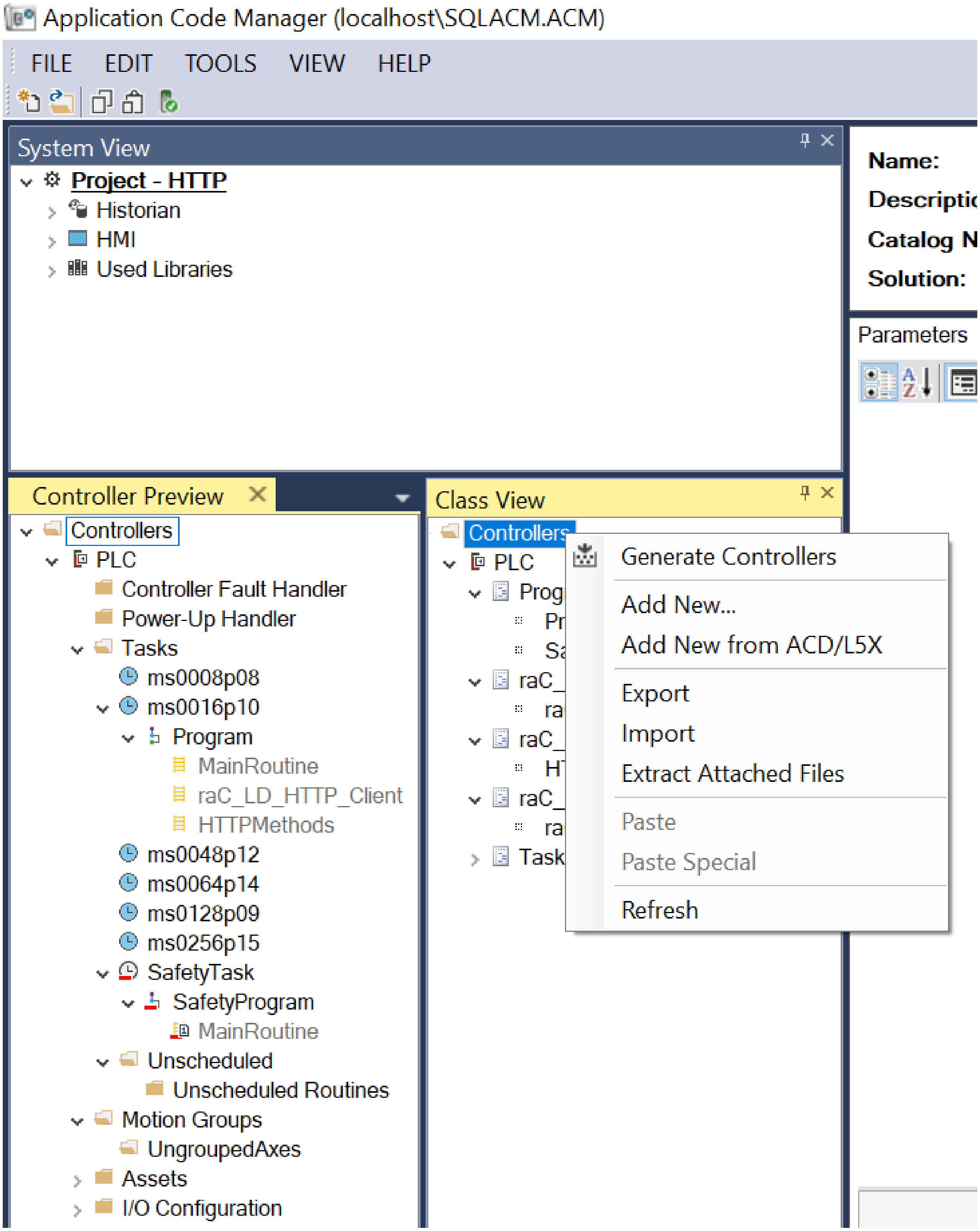
GETAcceptContentType
Specify the Accept Header of the request. Leave empty to use 'Accept: */*'

Click on next and finish

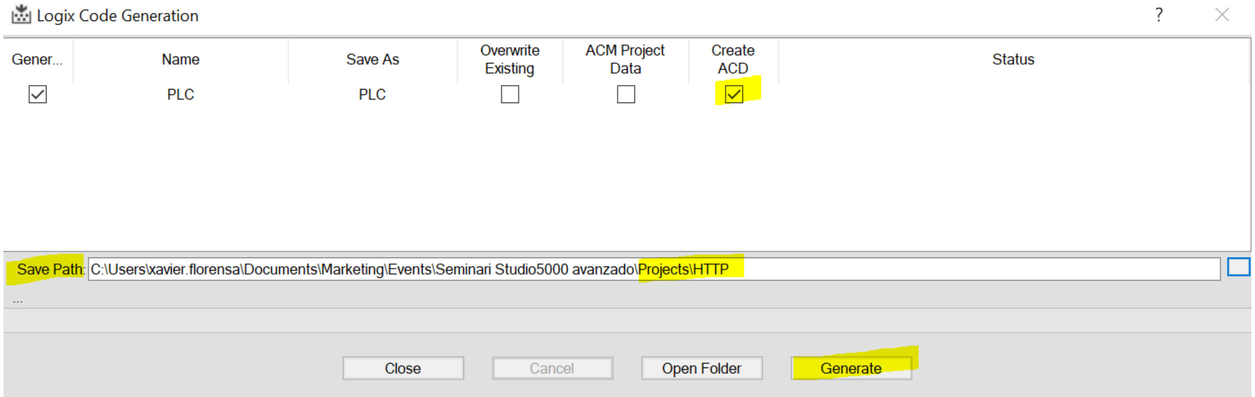

Application Code Manager (localhost\SQLACM.ACM)
FILE
EDIT
TOOLS
VIEW
HELP
System View
Project - HTTP
Historian
HMI
Used Libraries
Controller Preview
Controllers
PLC
Controller Fault Handler
Power-Up Handler
Tasks
ms0008p08
ms0016p10
Program
MainRoutine
raC_LD_HTTP_Client
HTTPMethods
ms0048p12
ms0064p14
ms0128p09
ms0256p15
SafetyTask
SafetyProgram
MainRoutine
Unscheduled
Unscheduled Routines
Motion Groups
UngroupedAxes
Assets
I/O Configuration
Class View
Controllers
PLC
Program (1.0)
Program
SafetyProgram
raC_LD_HTTP_Client (1.0)
raC_LD_HTTP_Client
raC_LD_HTTP_Methods (1.0)
HTTPMethods
raC_Opr_HTTP (1.0)
raC_Opr_HTTP
Task (1.0)

Generate the ACD file

With Generate controllers



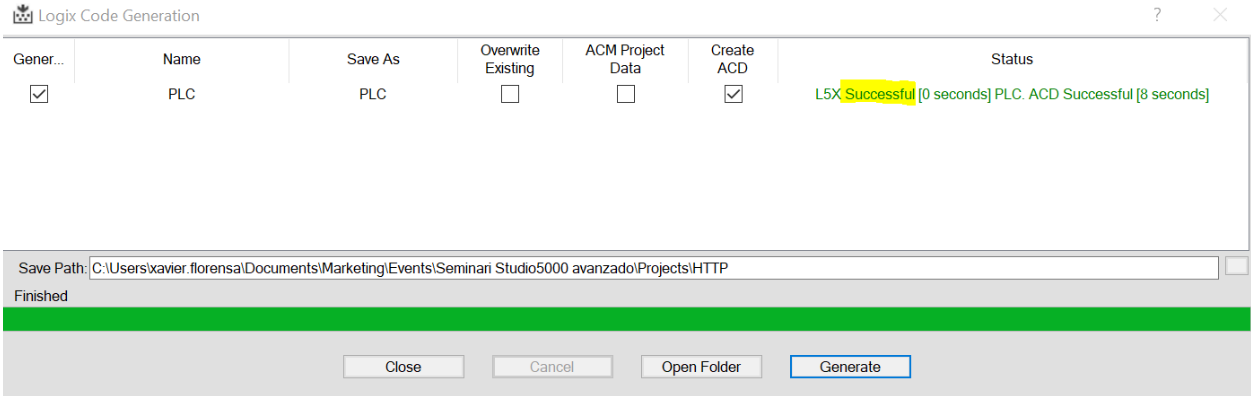
Save it



Gener...	Name	Save As	Overwrite Existing	ACM Project Data	Create ACD	Status
☑	PLC	PLC	□	□	☑	

Save Path: C:\Users\xavier.florensa\Documents\Marketing\Events\Seminari Studio5000 avanzado\Projects\HTTP

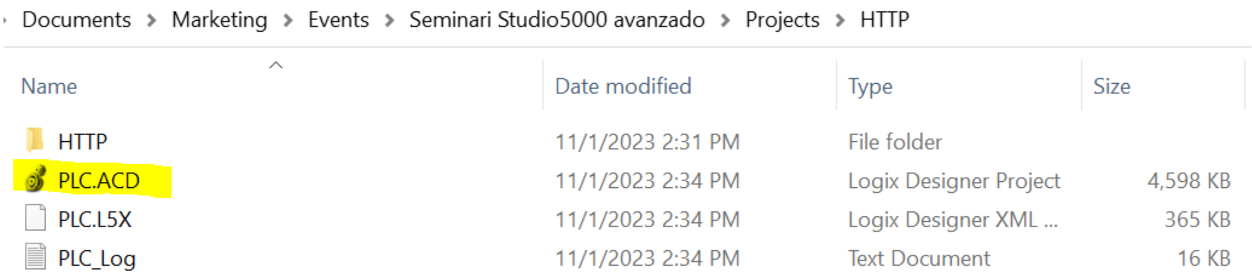
...



Gener...	Name	Save As	Overwrite Existing	ACM Project Data	Create ACD	Status
☑	PLC	PLC	□	□	☑	L5X Successful [0 seconds] PLC. ACD Successful [8 seconds]

Save Path: C:\Users\xavier.florensa\Documents\Marketing\Events\Seminari Studio5000 avanzado\Projects\HTTP

Finished



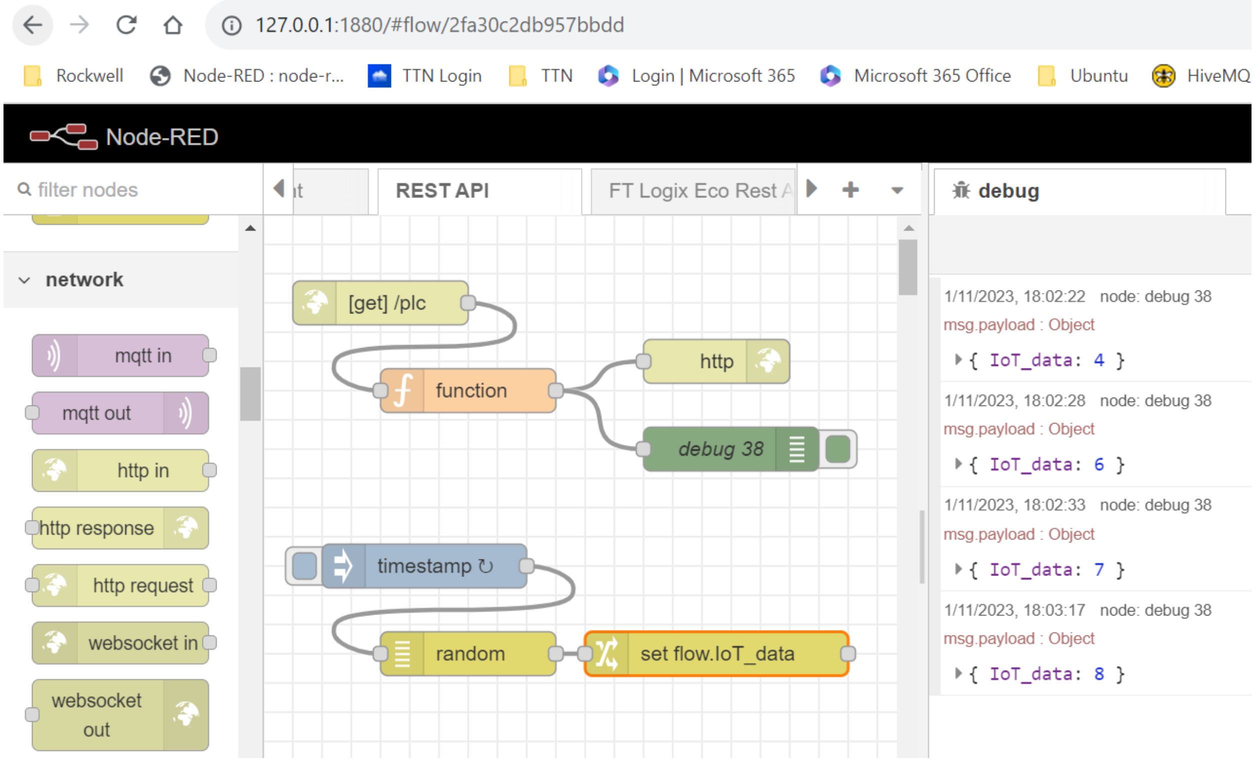
› Documents › Marketing › Events › Seminari Studio5000 avanzado › Projects › HTTP

Name	Date modified	Type	Size
HTTP	11/1/2023 2:31 PM	File folder	
PLC.ACD	11/1/2023 2:34 PM	Logix Designer Project	4,598 KB
PLC.L5X	11/1/2023 2:34 PM	Logix Designer XML ...	365 KB
PLC_Log	11/1/2023 2:34 PM	Text Document	16 KB

and open it on Studio5000.

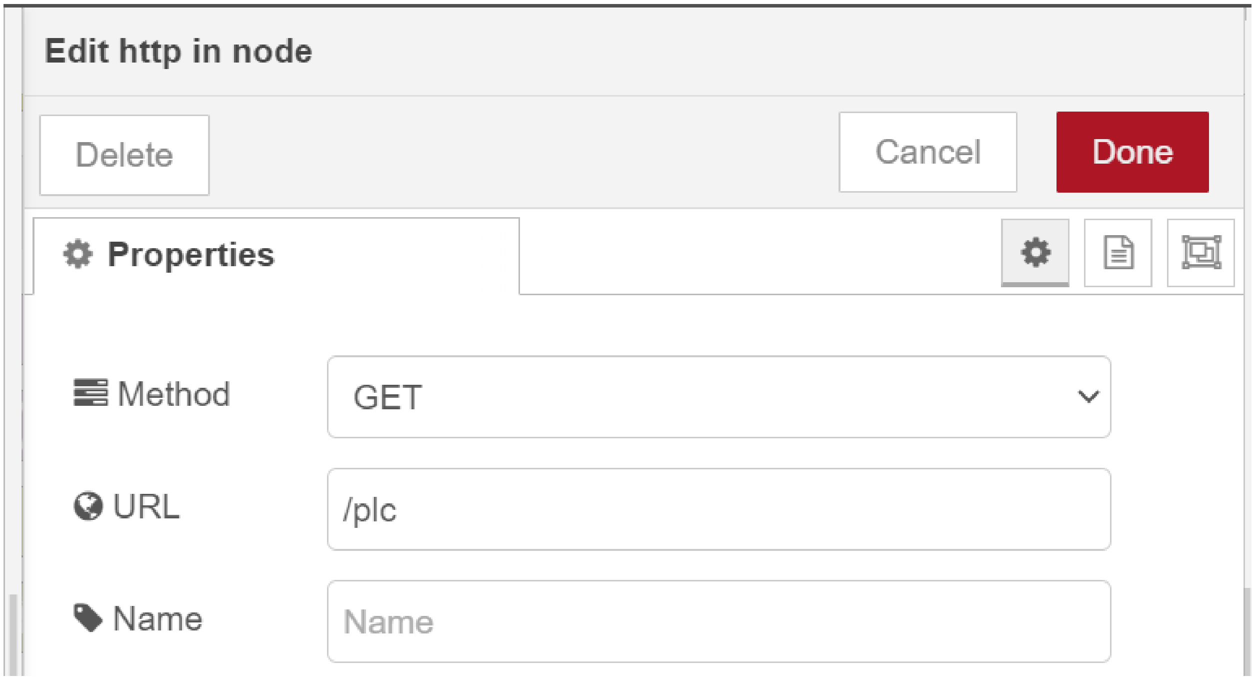
To test it let’s prepare a simple RESTFul API server, for instance one who returns the random value between 1 and 10.

This way with node-RED



```
1/11/2023, 18:02:22   node: debug 38
msg.payload : Object
▸{ IoT_data: 4 }
1/11/2023, 18:02:28   node: debug 38
msg.payload : Object
▸{ IoT_data: 6 }
1/11/2023, 18:02:33   node: debug 38
msg.payload : Object
▸{ IoT_data: 7 }
1/11/2023, 18:03:17   node: debug 38
msg.payload : Object
▸{ IoT_data: 8 }
```

And this is how the nodes are configured

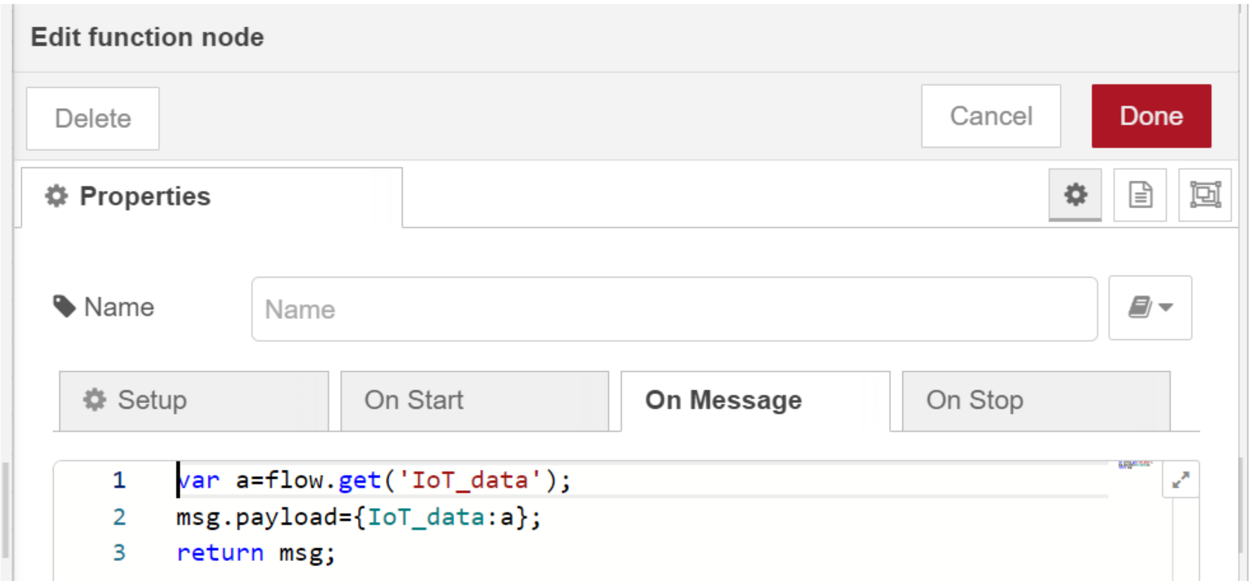


Edit http in node

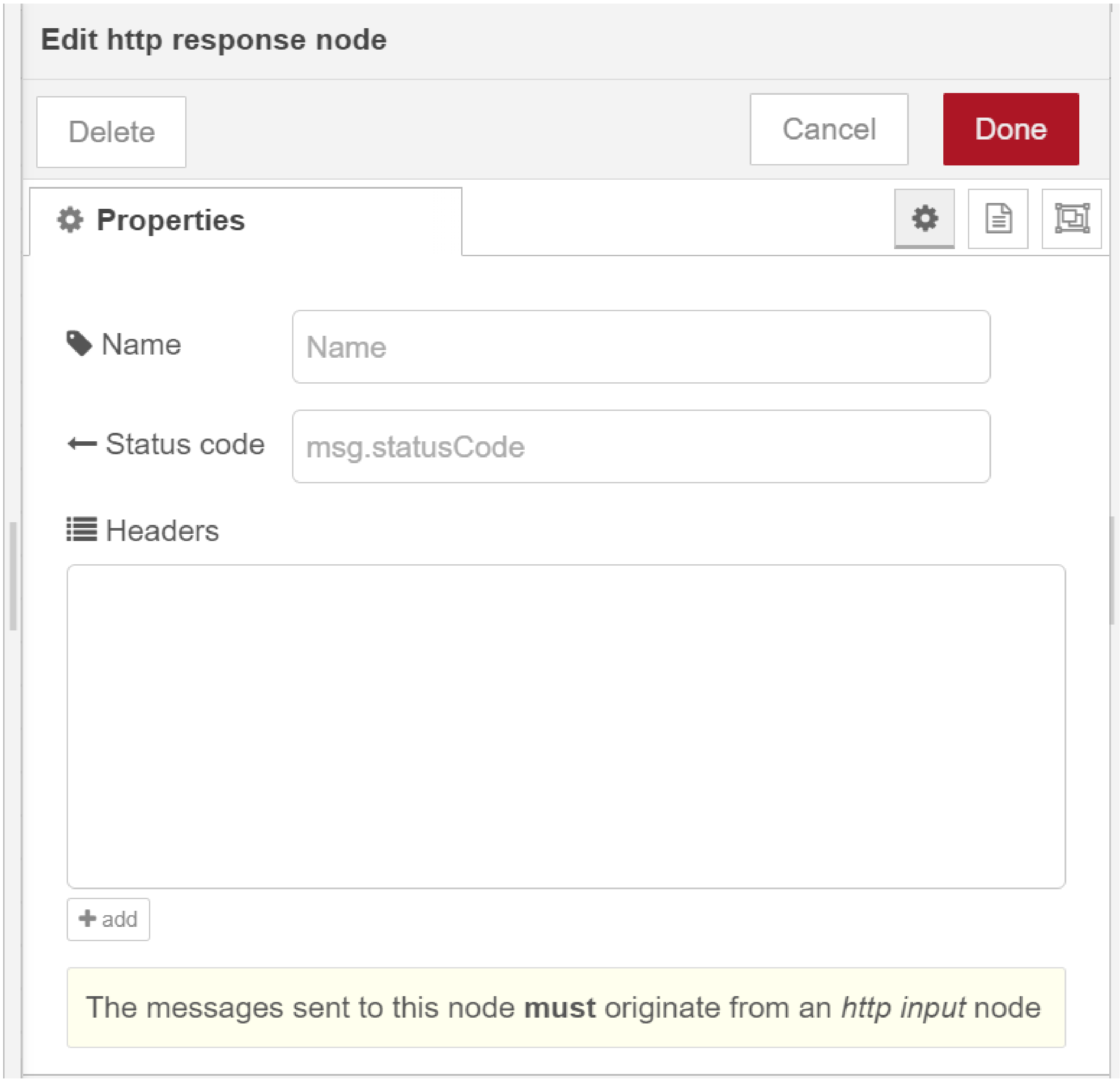
Method: GET

URL: /plc

Name: Name



```
var a=flow.get('IoT_data');
msg.payload={IoT_data:a};
return msg;
```



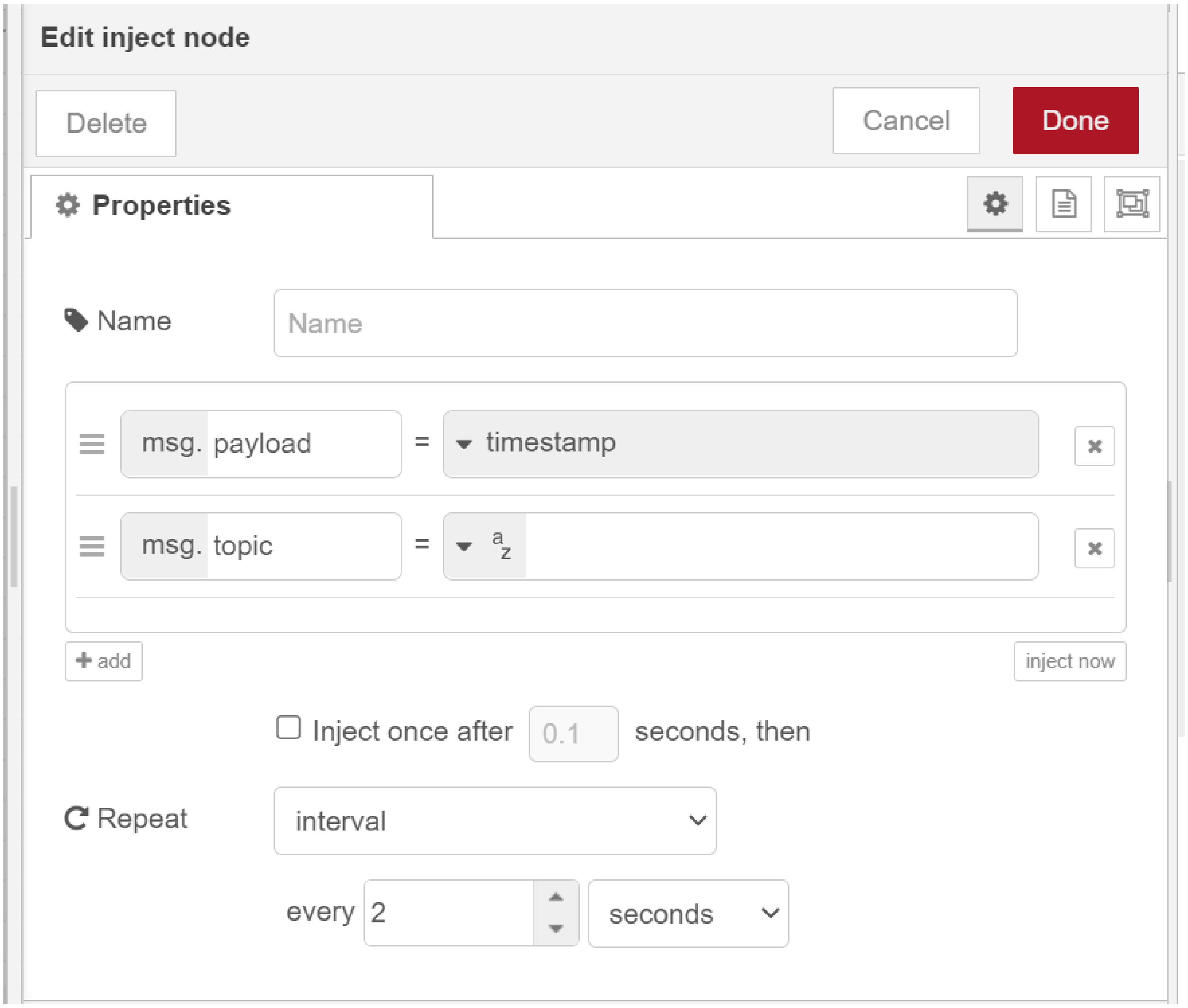
Edit http response node

Name: Name

Status code: msg.statusCode

Headers

The messages sent to this node **must** originate from an *http input* node



Name: Name

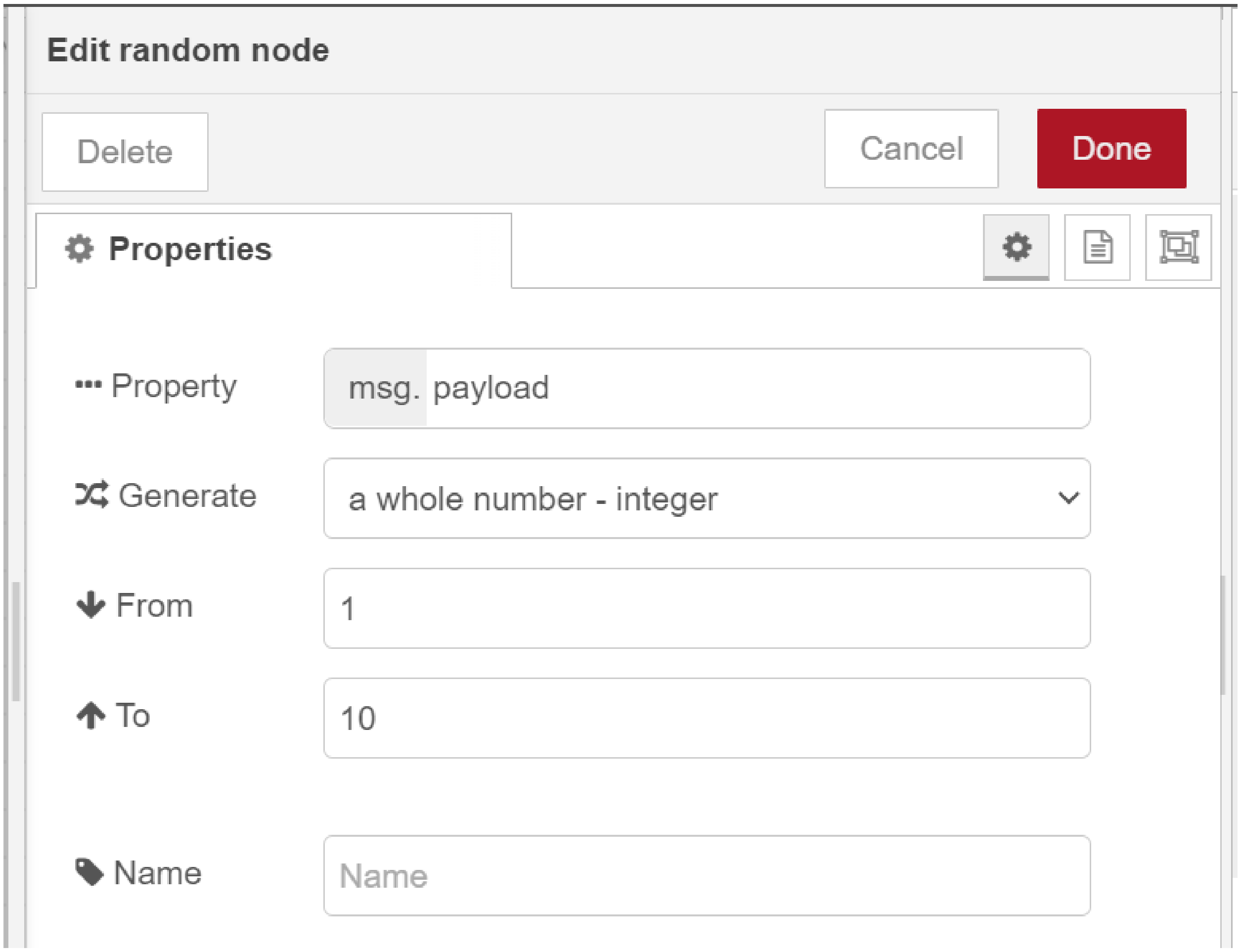
msg. payload = timestamp

msg. topic = $^{a}_{z}$

☐ Inject once after 0.1 seconds, then

Repeat: interval

every 2 seconds



Edit random node

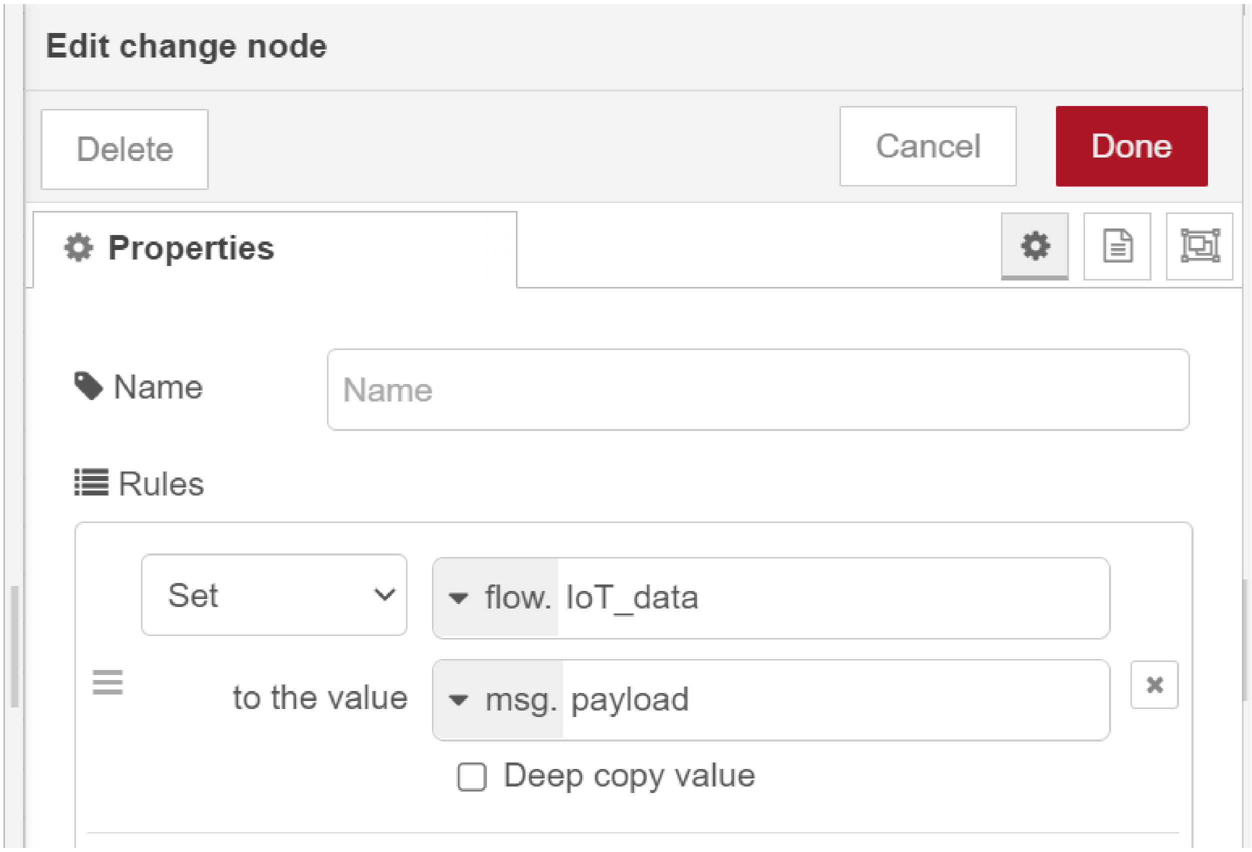
Property: msg. payload

Generate: a whole number - integer

From: 1

To: 10

Name: Name



Edit change node

Name: Name

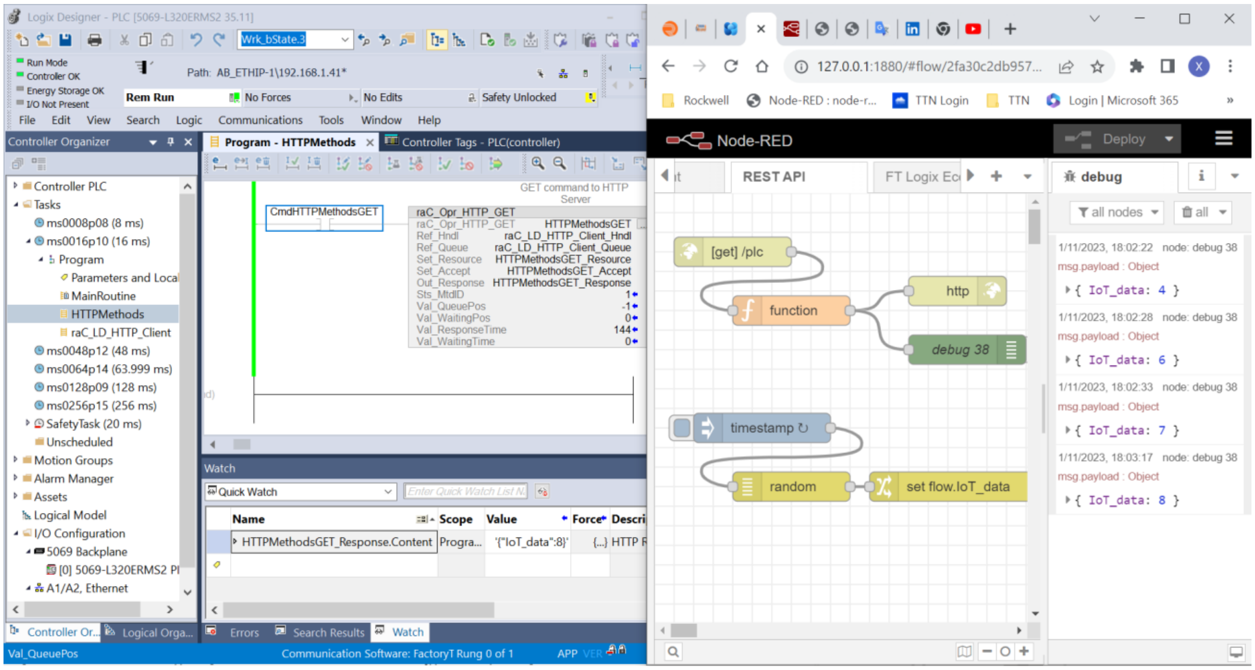
Rules

Set flow. IoT_data

to the value msg. payload

☐ Deep copy value

Voilà



Path: AB_ETHIP-1\192.168.1.41*

GET command to HTTP Server

CmdHTTPMethodsGET

raC_Opr_HTTP_GET	
raC_Opr_HTTP_GET	HTTPMethodsGET
Ref_Hndl	raC_LD_HTTP_Client_Hndl
Ref_Queue	raC_LD_HTTP_Client_Queue
Set_Resource	HTTPMethodsGET_Resource
Set_Accept	HTTPMethodsGET_Accept
Out_Response	HTTPMethodsGET_Response
Sts_MtdID	1
Val_QueuePos	-1
Val_WaitingPos	0
Val_ResponseTime	144
Val_WaitingTime	0

Name	Scope	Value	Force	Descri
HTTPMethodsGET_Response.Content	Progra...	'{"IoT_data":8}'	{...}	HTTP R

Val_QueuePos

Communication Software: FactoryT Rung 0 of 1

127.0.0.1:1880/#flow/2fa30c2db957...

```
1/11/2023, 18:02:22   node: debug 38
msg.payload : Object
▸{ IoT_data: 4 }
1/11/2023, 18:02:28   node: debug 38
msg.payload : Object
▸{ IoT_data: 6 }
1/11/2023, 18:02:33   node: debug 38
msg.payload : Object
▸{ IoT_data: 7 }
1/11/2023, 18:03:17   node: debug 38
msg.payload : Object
▸{ IoT_data: 8 }
```

As you can see on this video

https://youtu.be/UafslB52IMU

next, we want to Inject on InfluxdB directly from PLC

1.1. Installing InfluxDB on windows

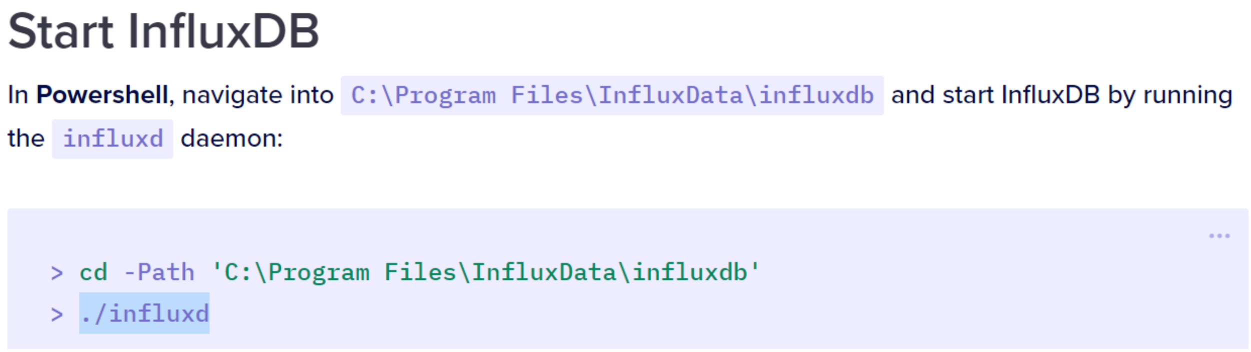
Using PowerShell in Administrator mode

Download with this command

```
wget https://dl.influxdata.com/influxdb/releases/influxdb2-2.7.3-windows.zip -UseBasicParsing -OutFile influxdb2-2.7.3-windows.zip

Expand-Archive .\influxdb2-2.7.3-windows.zip -DestinationPath 'C:\Program Files\InfluxData\influxdb\'
```

From this guide

https://portal.influxdata.com/downloads/



Start InfluxDB

In **Powershell**, navigate into `C:\Program Files\InfluxData\influxdb` and start InfluxDB by running the `influxd` daemon:

```
> cd -Path 'C:\Program Files\InfluxData\influxdb'
> ./influxd
```

```
> cd -Path 'C:\Program Files\InfluxData\influxdb'
> ./influxd
```

cd -Path 'C:\Program Files\InfluxData\influxdb'

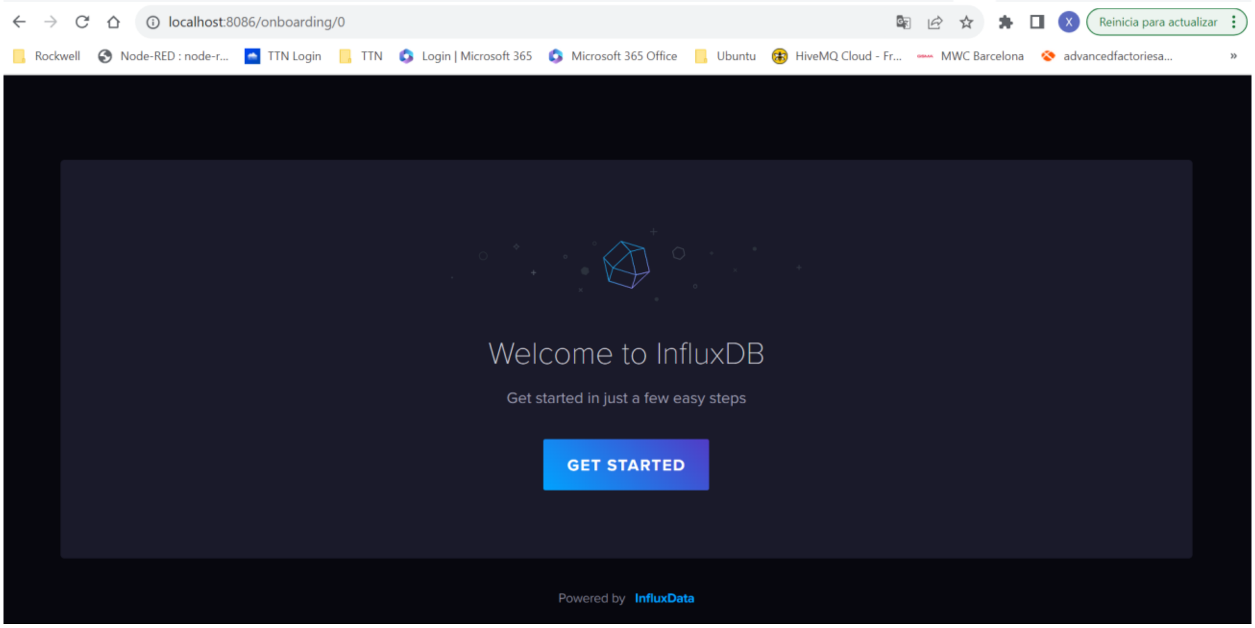
./influxd

Open an explorer with localhost and port 8086

Username

xavier

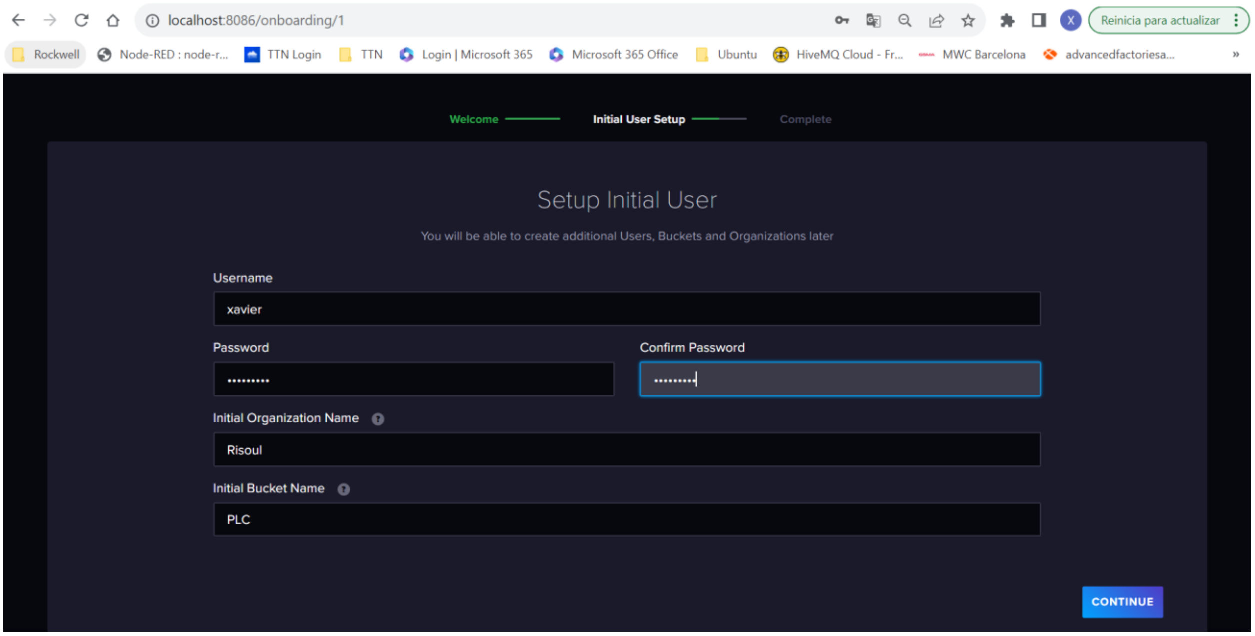
Password



Welcome to InfluxDB

Get started in just a few easy steps

Start building your influxdb use cases



Setup Initial User

You will be able to create additional Users, Buckets and Organizations later

Username: xavier

Password: •••••••••

Confirm Password: •••••••••

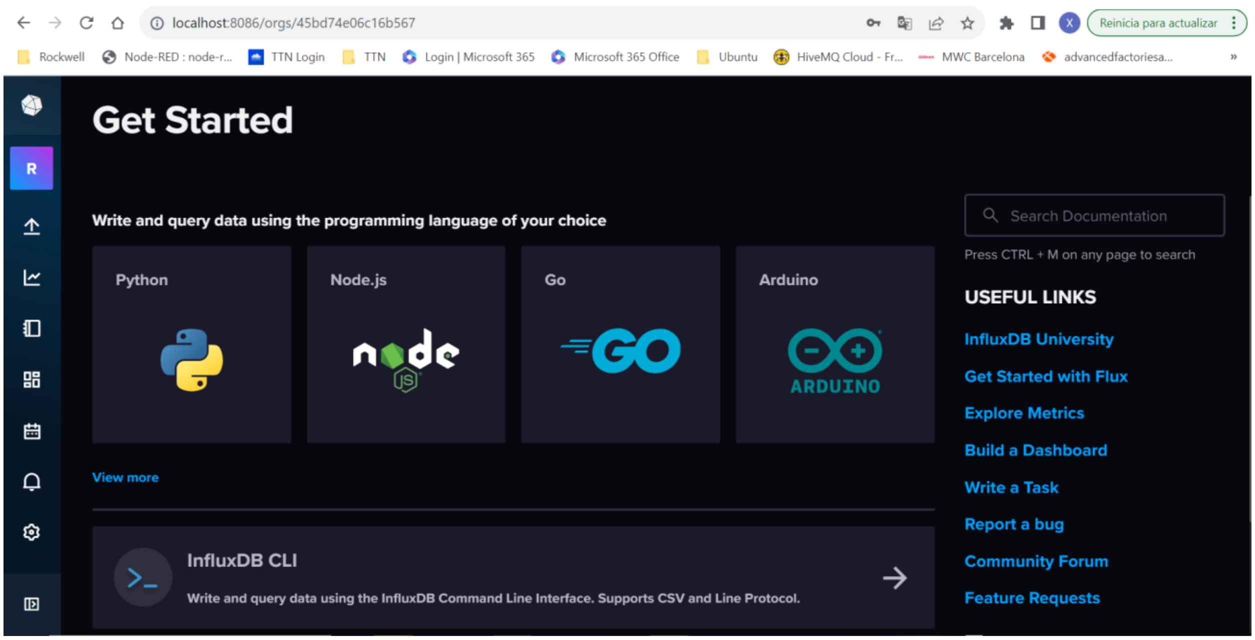
Initial Organization Name: Risoul

Initial Bucket Name: PLC

API tocken

MwQvFkNF8uI_Yx8327ohwgDG2qHBhO9ZbAqbpFPcFRX6amE9SooSyxAiA9zofuxj8c_C26cf-zGmMeLyGYKgHA==

Select Quickstart



Get Started

Write and query data using the programming language of your choice

View more

Press CTRL + M on any page to search

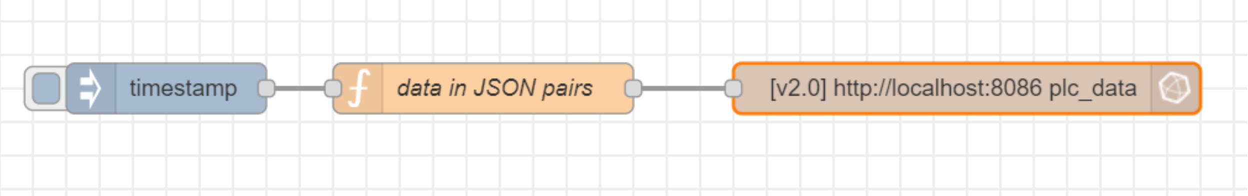
USEFUL LINKS

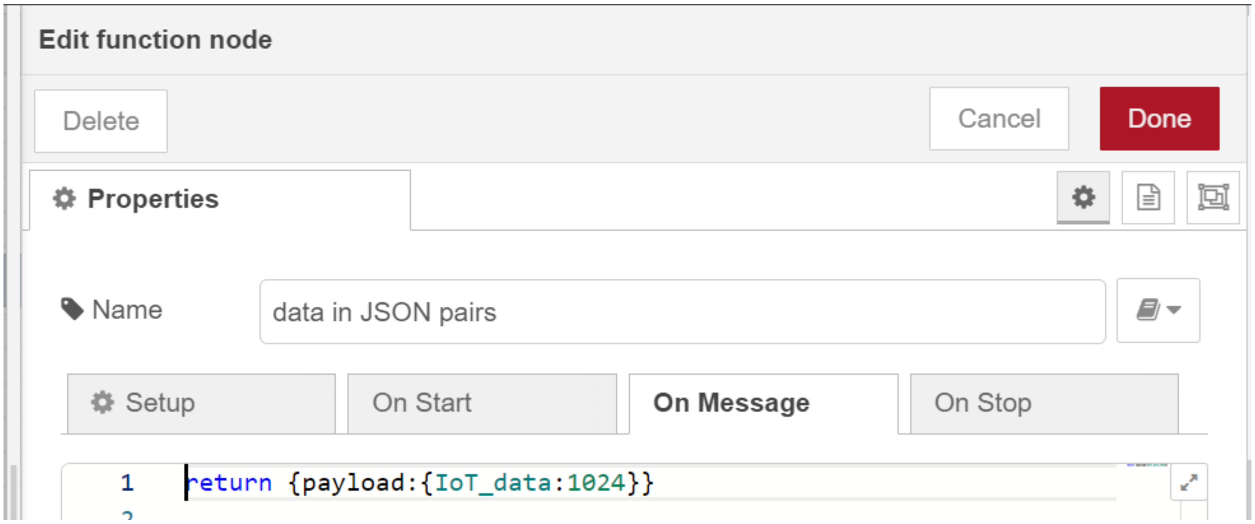
InfluxDB University
Get Started with Flux
Explore Metrics
Build a Dashboard
Write a Task
Report a bug
Community Forum
Feature Requests

InfluxDB CLI

Write and query data using the InfluxDB Command Line Interface. Supports CSV and Line Protocol.

1.2. Inject your first data with node-red

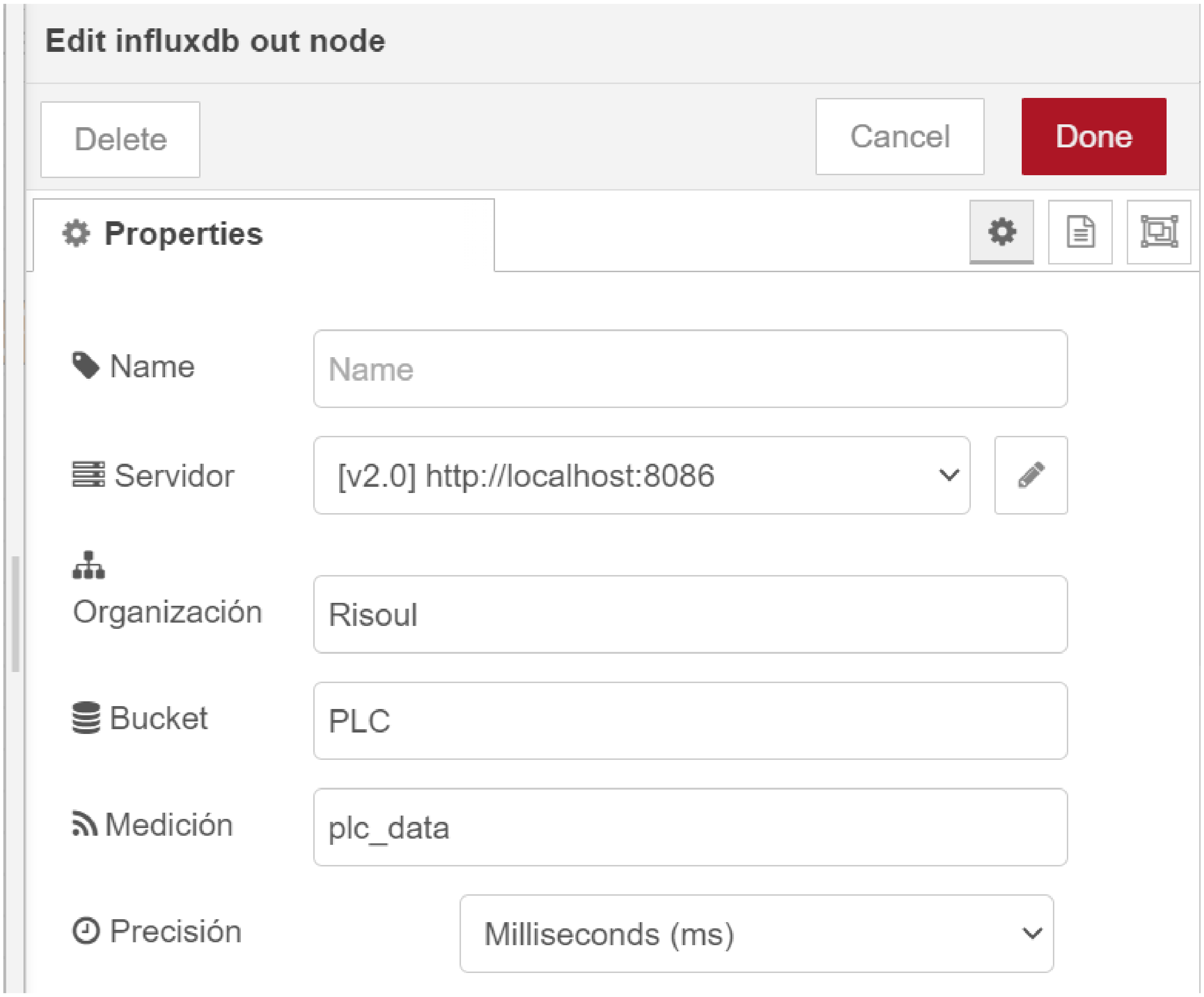




Name: data in JSON pairs

```
return {payload:{IoT_data:1024}}
```

Let’s try to inject to INfluxDB using an Http request



Edit influxdb out node

Name: Name

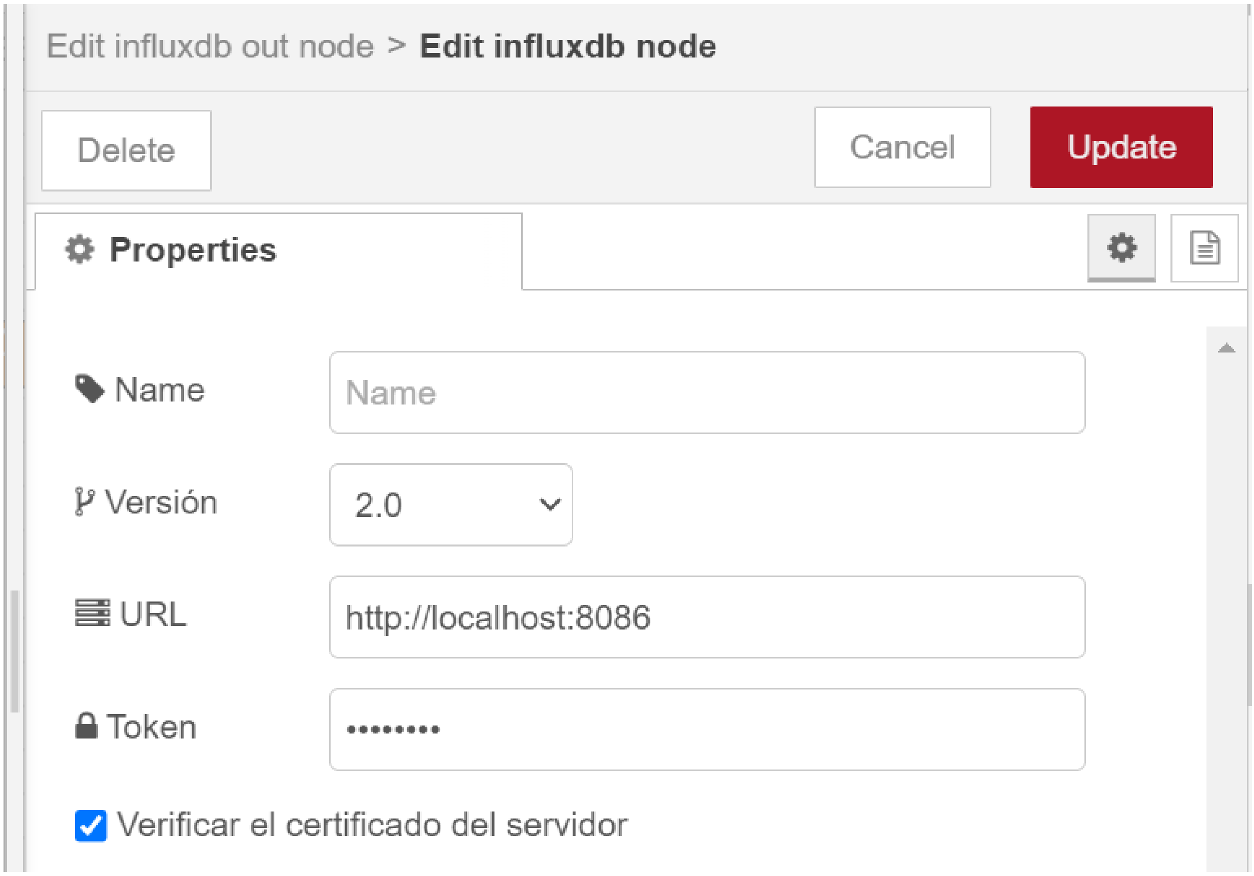
Servidor: [v2.0] http://localhost:8086

Organización: Risoul

Bucket: PLC

Medición: plc_data

Precisión: Milliseconds (ms)



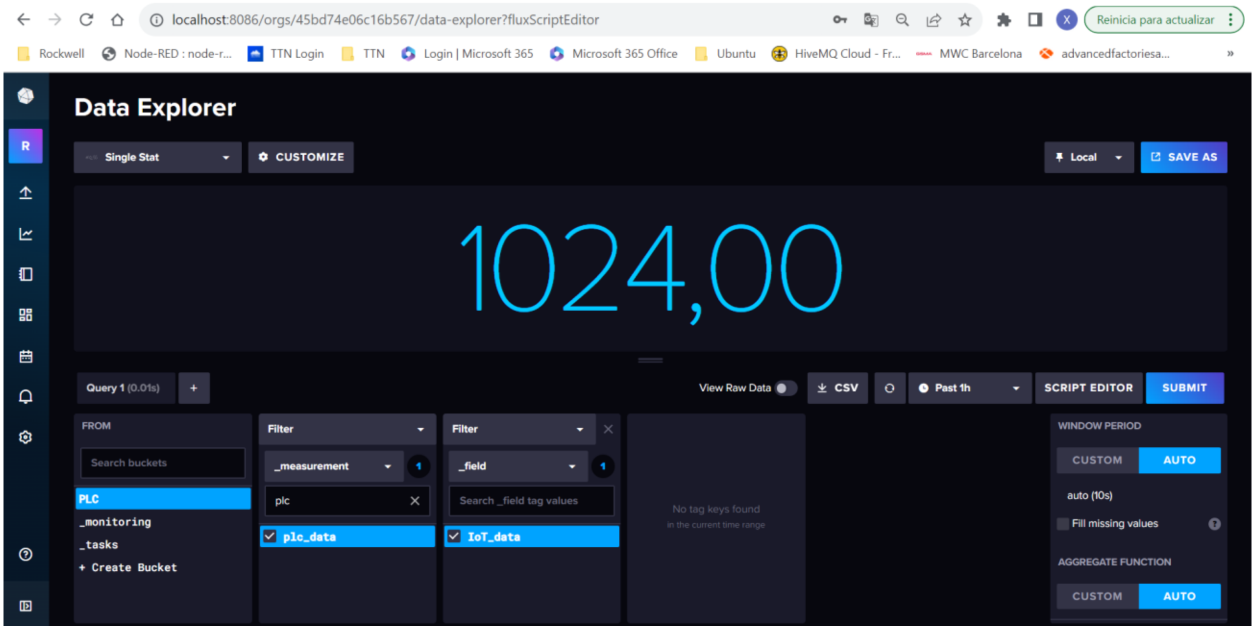
Name: Name

Versión: 2.0

URL: http://localhost:8086

Token: •••••••••

☑ Verificar el certificado del servidor



1024,00

FROM: Search buckets — PLC, _monitoring, _tasks, + Create Bucket

Filter: _measurement 1 — plc — ☑ plc_data

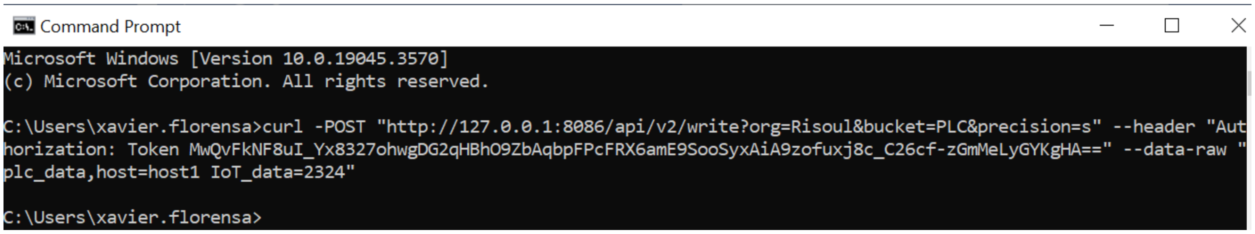
Filter: _field 1 — Search _field tag values — ☑ IoT_data

No tag keys found in the current time range

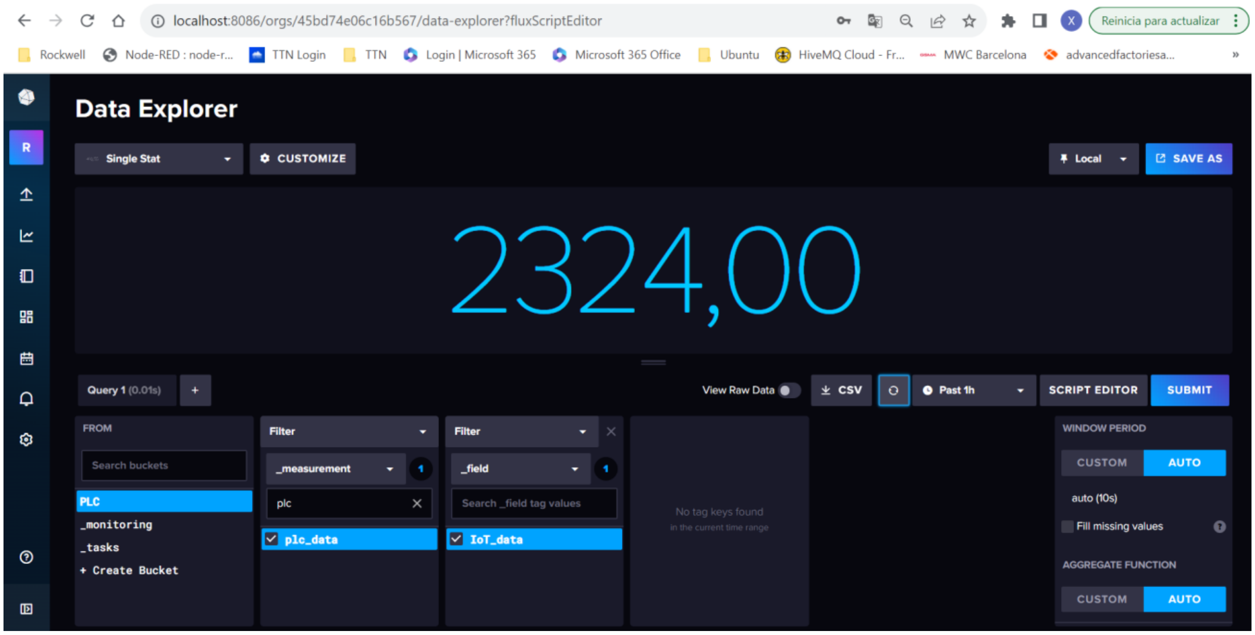
WINDOW PERIOD: CUSTOM AUTO — auto (10s) — □ Fill missing values

AGGREGATE FUNCTION: CUSTOM AUTO

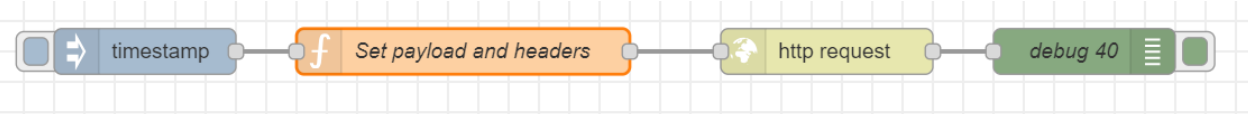
Now let’s try to build the http request ourselves curl -POST "http://127.0.0.1:8086/api/v2/write?org=Risoul&bucket=PLC&precision=s" --header "Authorization: Token MwQvFkNF8uI_Yx8327ohwgDG2qHBhO9ZbAqbpFPcFRX6amE9SooSyxAiA9zofuxj8c_C26cf-zGmMeLyGYKgHA==" --data-raw "plc_data,host=host1 IoT_data=2324"

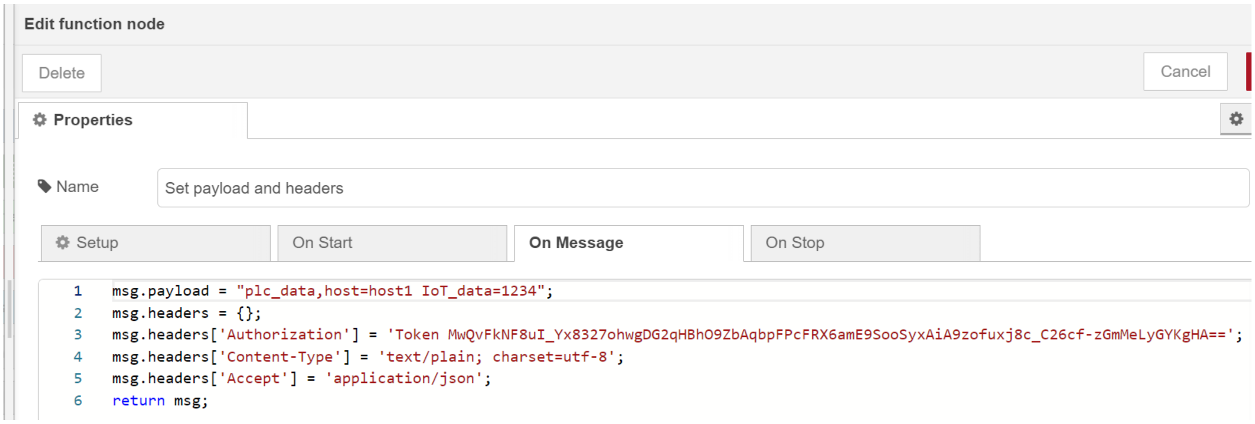


```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\xavier.florensa>curl -POST "http://127.0.0.1:8086/api/v2/write?org=Risoul&bucket=PLC&precision=s" --header "Aut
horization: Token MwQvFkNF8uI_Yx8327ohwgDG2qHBhO9ZbAqbpFPcFRX6amE9SooSyxAiA9zofuxj8c_C26cf-zGmMeLyGYKgHA==" --data-raw "
plc_data,host=host1 IoT_data=2324"

C:\Users\xavier.florensa>
```



Let's do this from Node-RED

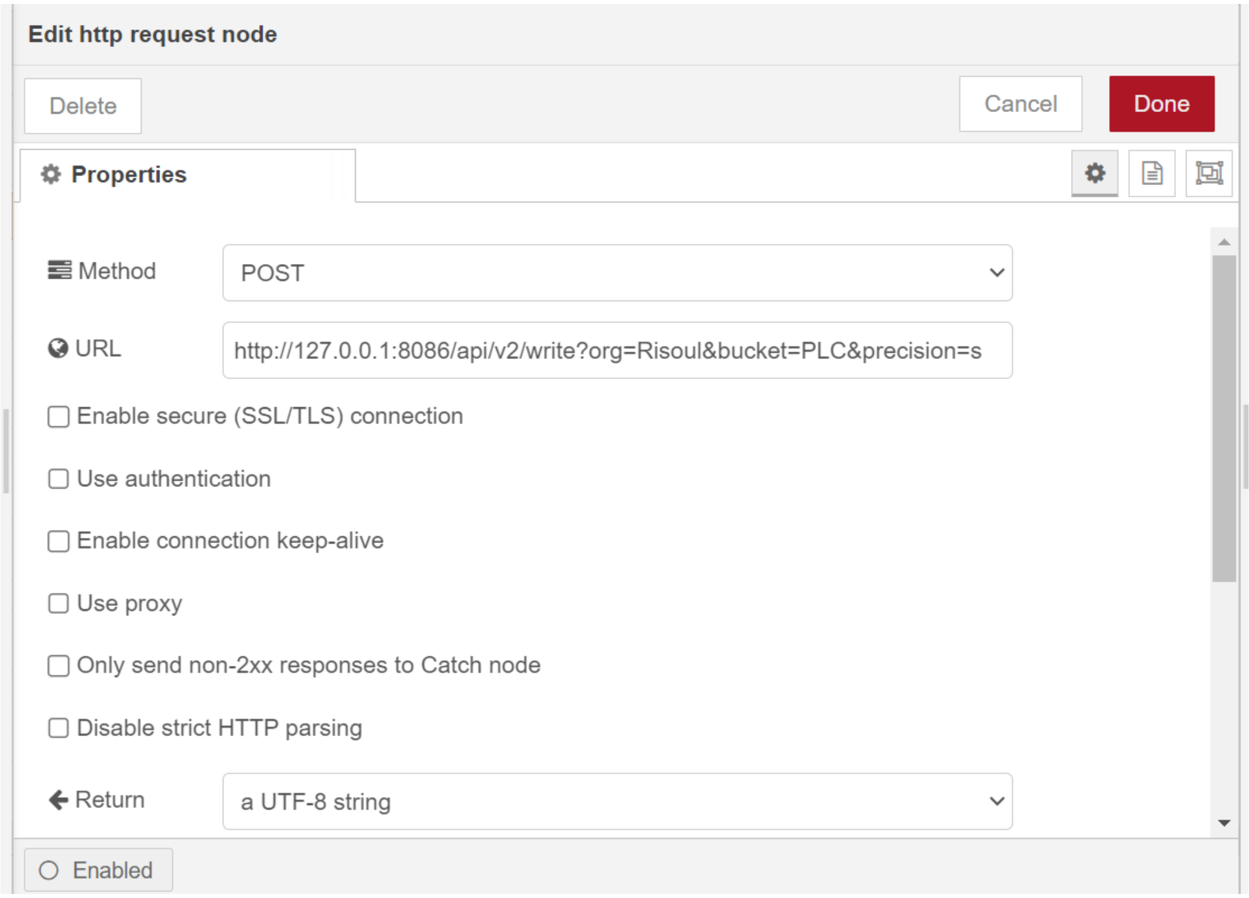




Name: Set payload and headers

```
msg.payload = "plc_data,host=host1 IoT_data=1234";
msg.headers = {};
msg.headers['Authorization'] = 'Token MwQvFkNF8uI_Yx8327ohwgDG2qHBhO9ZbAqbpFPcFRX6amE9SooSyxAiA9zofuxj8c_C26cf-zGmMeLyGYKgHA==';
msg.headers['Content-Type'] = 'text/plain; charset=utf-8';
msg.headers['Accept'] = 'application/json';
return msg;
```

```
msg.payload = "plc_data,host=host1 IoT_data=1234";
msg.headers = {};
msg.headers['Authorization'] = 'Token MwQvFkNF8uI_Yx8327ohwgDG2qHBhO9ZbAqbpFPcFRX6amE9SooSyxAiA9zofuxj8c_C26cf-zGmMeLyGYKgHA==';
msg.headers['Content-Type'] = 'text/plain; charset=utf-8';
msg.headers['Accept'] = 'application/json';
return msg;
```



Method: POST

URL: http://127.0.0.1:8086/api/v2/write?org=Risoul&bucket=PLC&precision=s

☐ Enable secure (SSL/TLS) connection

☐ Use authentication

☐ Enable connection keep-alive

☐ Use proxy

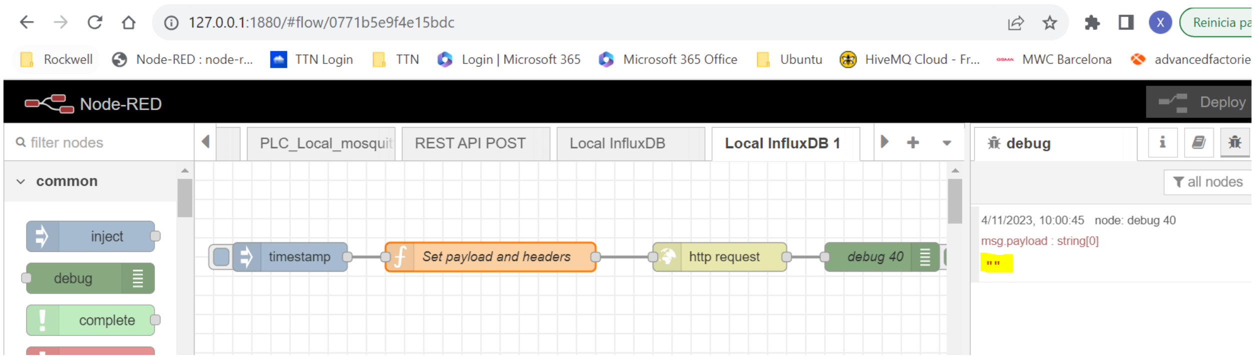
☐ Only send non-2xx responses to Catch node

☐ Disable strict HTTP parsing

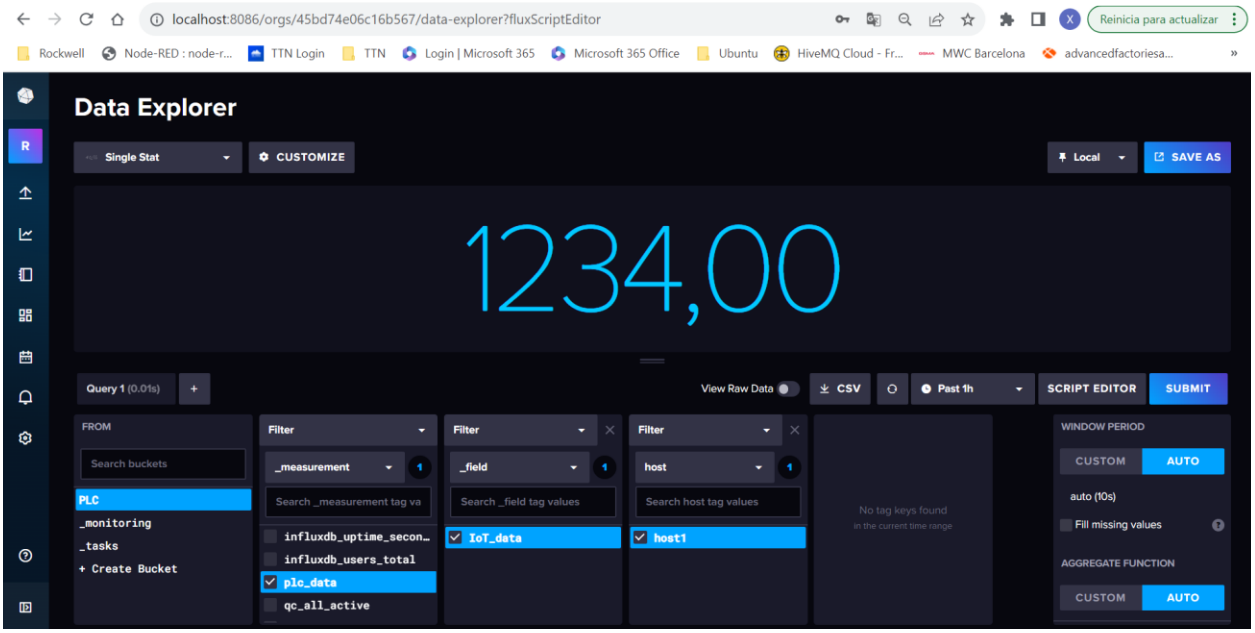
Return: a UTF-8 string

○ Enabled

Success



4/11/2023, 10:00:45 node: debug 40
msg.payload : string[0]
""



Now, let’s try to do this from our PLC

1.3. Injecting in InfluxDB directly from PLC

You can see the final result on this video

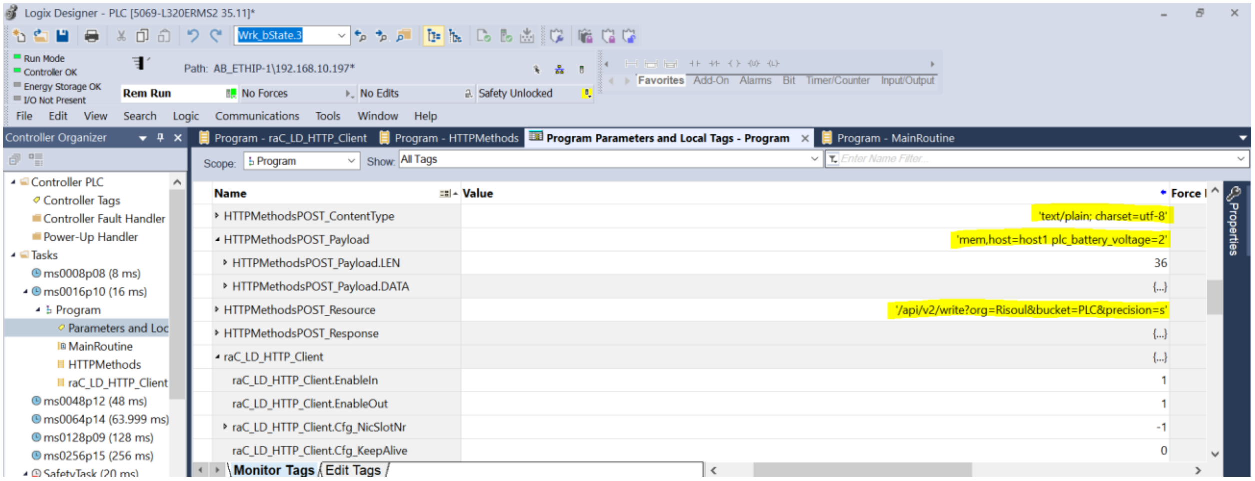
https://youtu.be/tDezJRUEjq4

The test was done with a PLC in DHCP connected to a wireless router thru patchcord cable

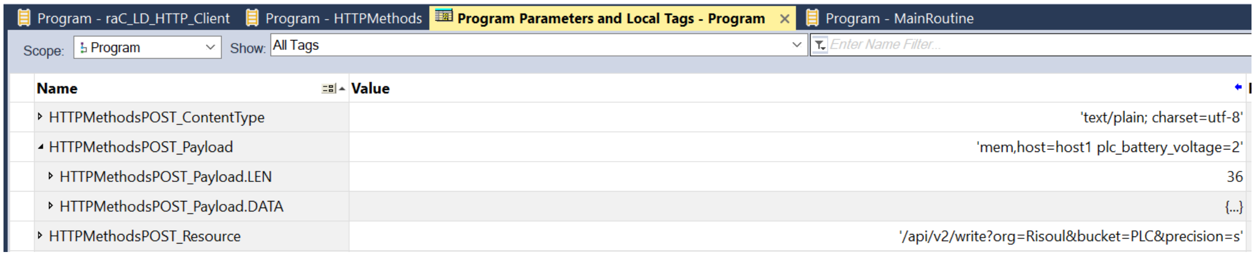
PLC has this address: 192.168.10.197

The computer has this Wi-fi address per DHCP: 192.168.10.105

Adjust these values on PLC

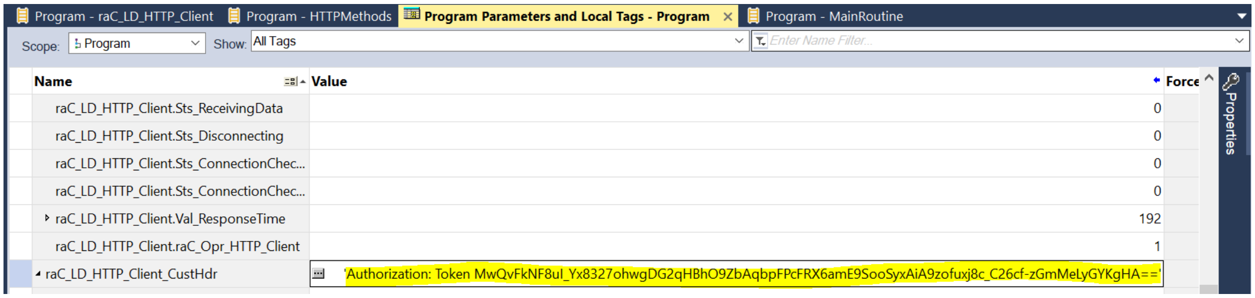


Name	Value	Force
HTTPMethodsPOST_ContentType	'text/plain; charset=utf-8'	
HTTPMethodsPOST_Payload	'mem,host=host1 plc_battery_voltage=2'	
HTTPMethodsPOST_Payload.LEN	36	
HTTPMethodsPOST_Payload.DATA	{...}	
HTTPMethodsPOST_Resource	'/api/v2/write?org=Risoul&bucket=PLC&precision=s'	
HTTPMethodsPOST_Response	{...}	
raC_LD_HTTP_Client	{...}	
raC_LD_HTTP_Client.EnableIn	1	
raC_LD_HTTP_Client.EnableOut	1	
raC_LD_HTTP_Client.Cfg_NicSlotNr	-1	
raC_LD_HTTP_Client.Cfg_KeepAlive	0	



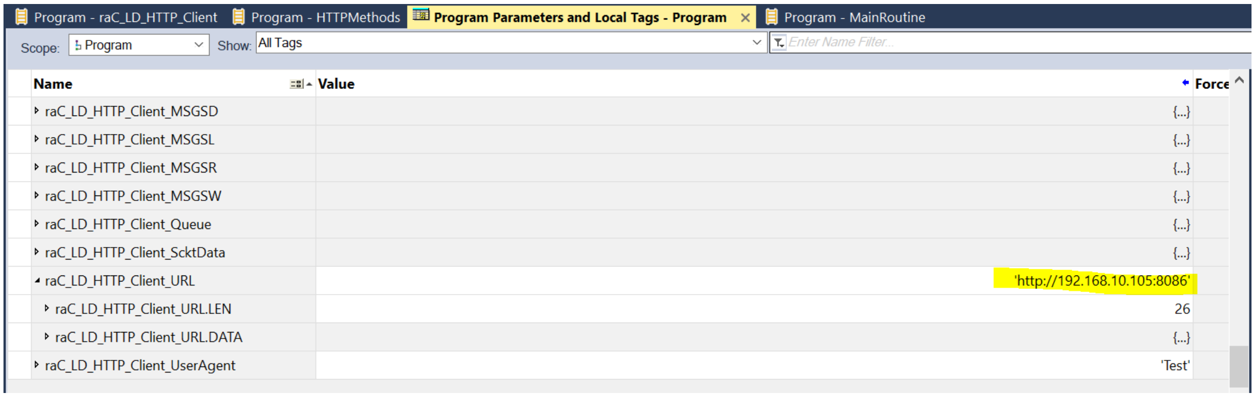
Name	Value
HTTPMethodsPOST_ContentType	'text/plain; charset=utf-8'
HTTPMethodsPOST_Payload	'mem,host=host1 plc_battery_voltage=2'
HTTPMethodsPOST_Payload.LEN	36
HTTPMethodsPOST_Payload.DATA	{...}
HTTPMethodsPOST_Resource	'/api/v2/write?org=Risoul&bucket=PLC&precision=s'

And the custom header



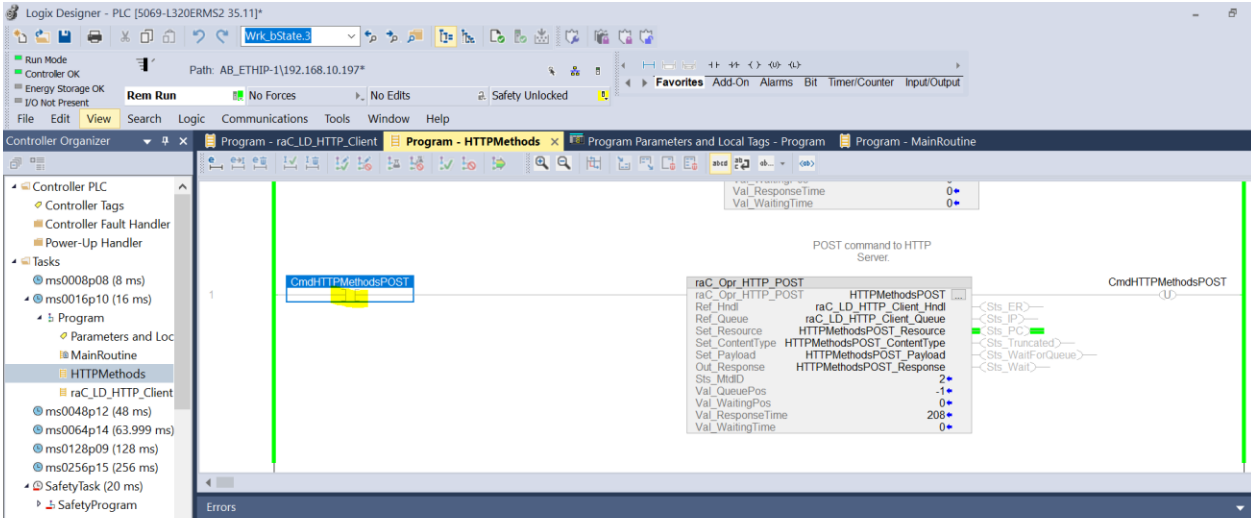
Name	Value	Force
raC_LD_HTTP_Client.Sts_ReceivingData	0	
raC_LD_HTTP_Client.Sts_Disconnecting	0	
raC_LD_HTTP_Client.Sts_ConnectionChec...	0	
raC_LD_HTTP_Client.Sts_ConnectionChec...	0	
raC_LD_HTTP_Client.Val_ResponseTime	192	
raC_LD_HTTP_Client.raC_Opr_HTTP_Client	1	
raC_LD_HTTP_Client_CustHdr	'Authorization: Token MwQvFkNF8ul_Yx8327ohwgDG2qHBhO9ZbAqbpFPcFRX6amE9SooSyxAiA9zofuxj8c_C26cf-zGmMeLyGYKgHA=='	

And the InfluxDB server IP



Name	Value	Force
raC_LD_HTTP_Client_MSGSD	{...}	
raC_LD_HTTP_Client_MSGSL	{...}	
raC_LD_HTTP_Client_MSGSR	{...}	
raC_LD_HTTP_Client_MSGSW	{...}	
raC_LD_HTTP_Client_Queue	{...}	
raC_LD_HTTP_Client_ScktData	{...}	
raC_LD_HTTP_Client_URL	'http://192.168.10.105:8086'	
raC_LD_HTTP_Client_URL.LEN	26	
raC_LD_HTTP_Client_URL.DATA	{...}	
raC_LD_HTTP_Client_UserAgent	'Test'	

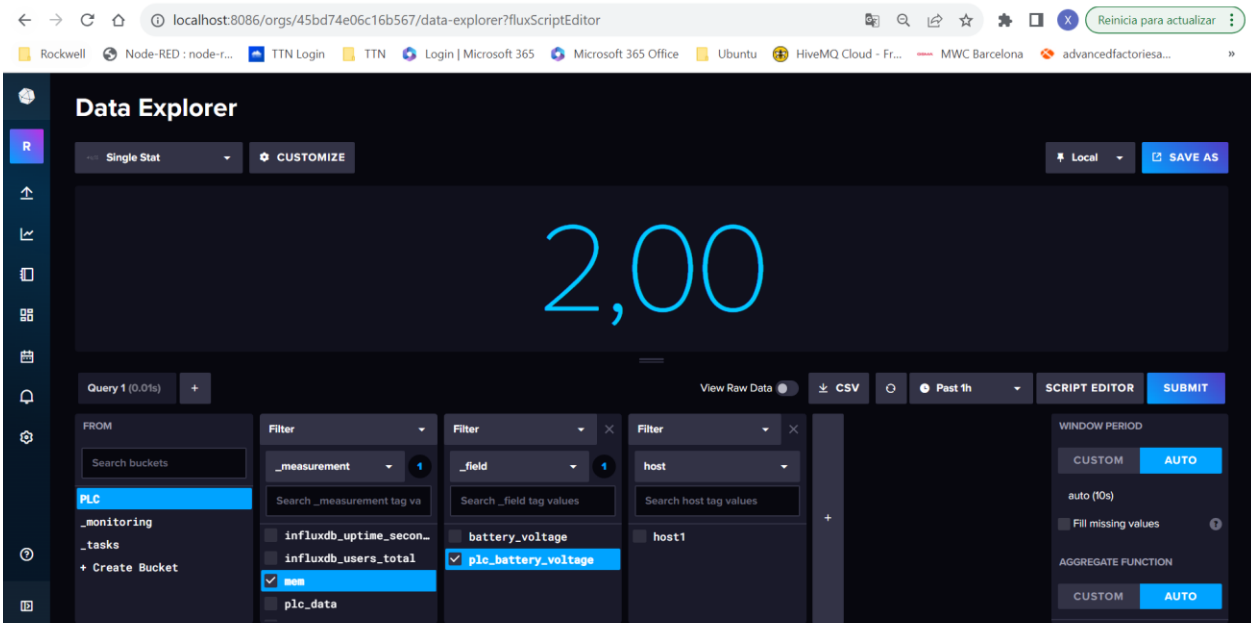
Toggle Post bit



raC_Opr_HTTP_POST	
Val_ResponseTime	0
Val_WaitingTime	0

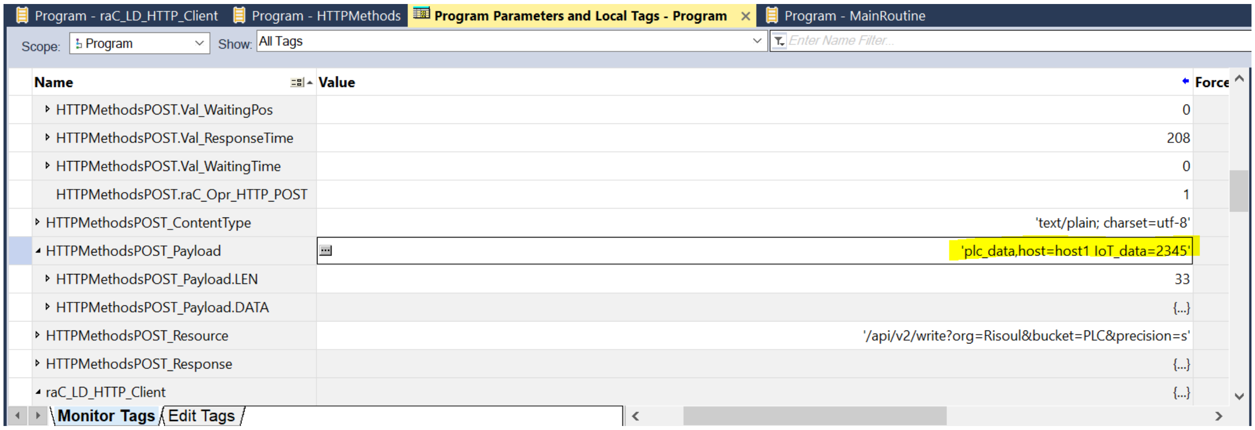
raC_Opr_HTTP_POST	
raC_Opr_HTTP_POST	HTTPMethodsPOST
Ref_Hndl	raC_LD_HTTP_Client_Hndl
Ref_Queue	raC_LD_HTTP_Client_Queue
Set_Resource	HTTPMethodsPOST_Resource
Set_ContentType	HTTPMethodsPOST_ContentType
Set_Payload	HTTPMethodsPOST_Payload
Out_Response	HTTPMethodsPOST_Response
Sts_MtdID	2
Val_QueuePos	-1
Val_WaitingPos	0
Val_ResponseTime	208
Val_WaitingTime	0

It works!!!!



localhost:8086/orgs/45bd74e06c16b567/data-explorer?fluxScriptEditor

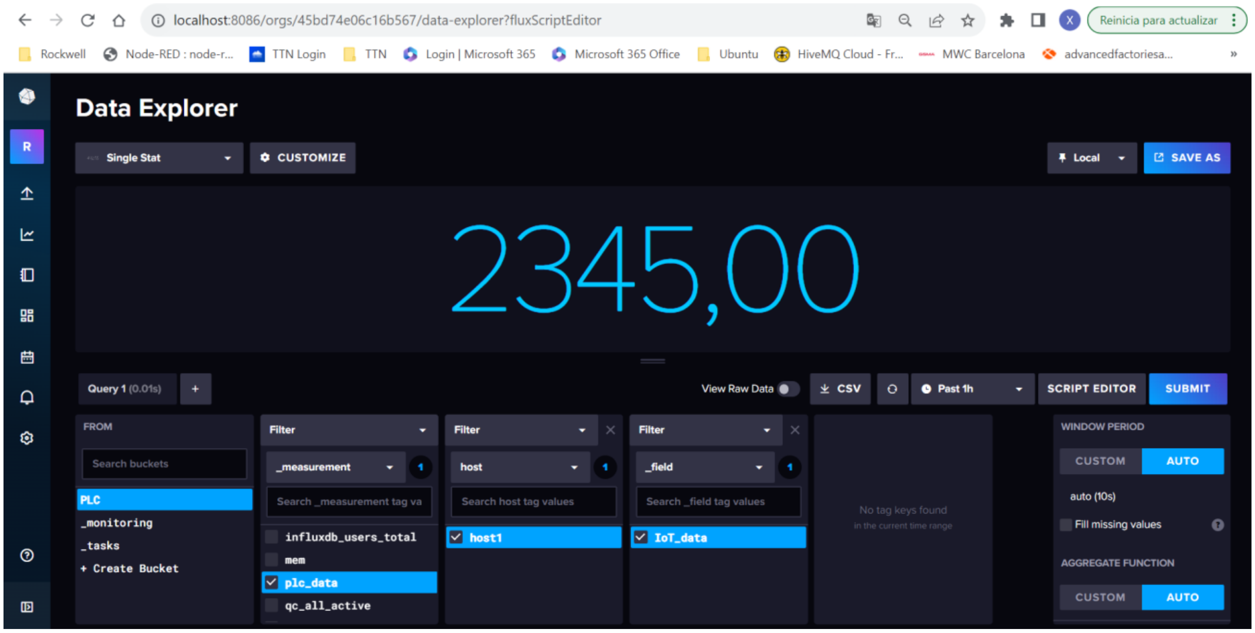
Now let's try with this value



Scope: Program Show: All Tags

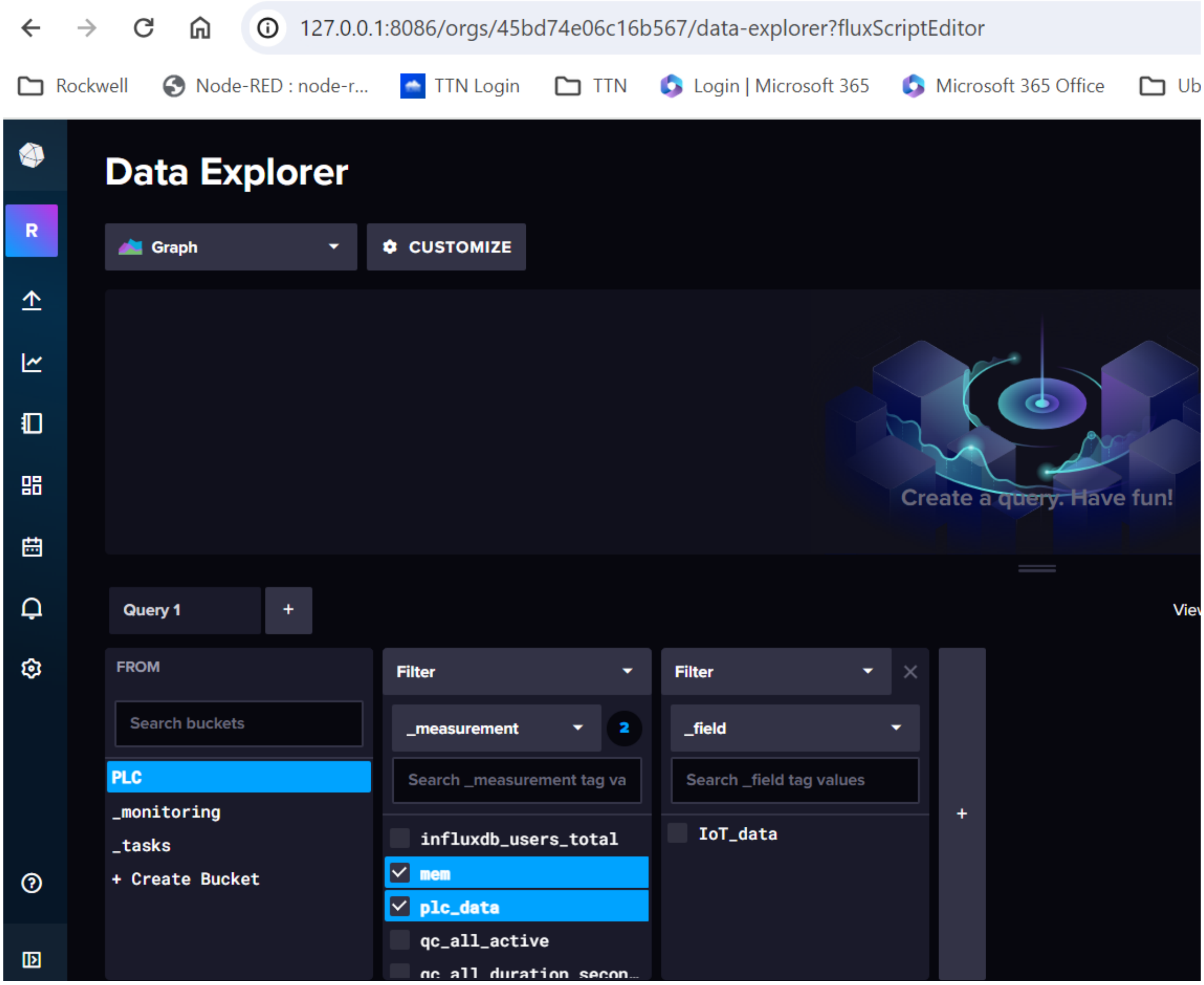
Name	Value	Force
HTTPMethodsPOST.Val_WaitingPos	0	
HTTPMethodsPOST.Val_ResponseTime	208	
HTTPMethodsPOST.Val_WaitingTime	0	
HTTPMethodsPOST.raC_Opr_HTTP_POST	1	
HTTPMethodsPOST_ContentType	'text/plain; charset=utf-8'	
HTTPMethodsPOST_Payload	'plc_data,host=host1 IoT_data=2345'	
HTTPMethodsPOST_Payload.LEN	33	
HTTPMethodsPOST_Payload.DATA	{...}	
HTTPMethodsPOST_Resource	'/api/v2/write?org=Risoul&bucket=PLC&precision=s'	
HTTPMethodsPOST_Response	{...}	
raC_LD_HTTP_Client	{...}	

Toggle Post bit



localhost:8086/orgs/45bd74e06c16b567/data-explorer?fluxScriptEditor

We can also try with this filtering



127.0.0.1:8086/orgs/45bd74e06c16b567/data-explorer?fluxScriptEditor

Data Explorer

FROM

Search buckets

- PLC
- _monitoring
- _tasks
- + Create Bucket

Filter

_measurement 2

Search _measurement tag va

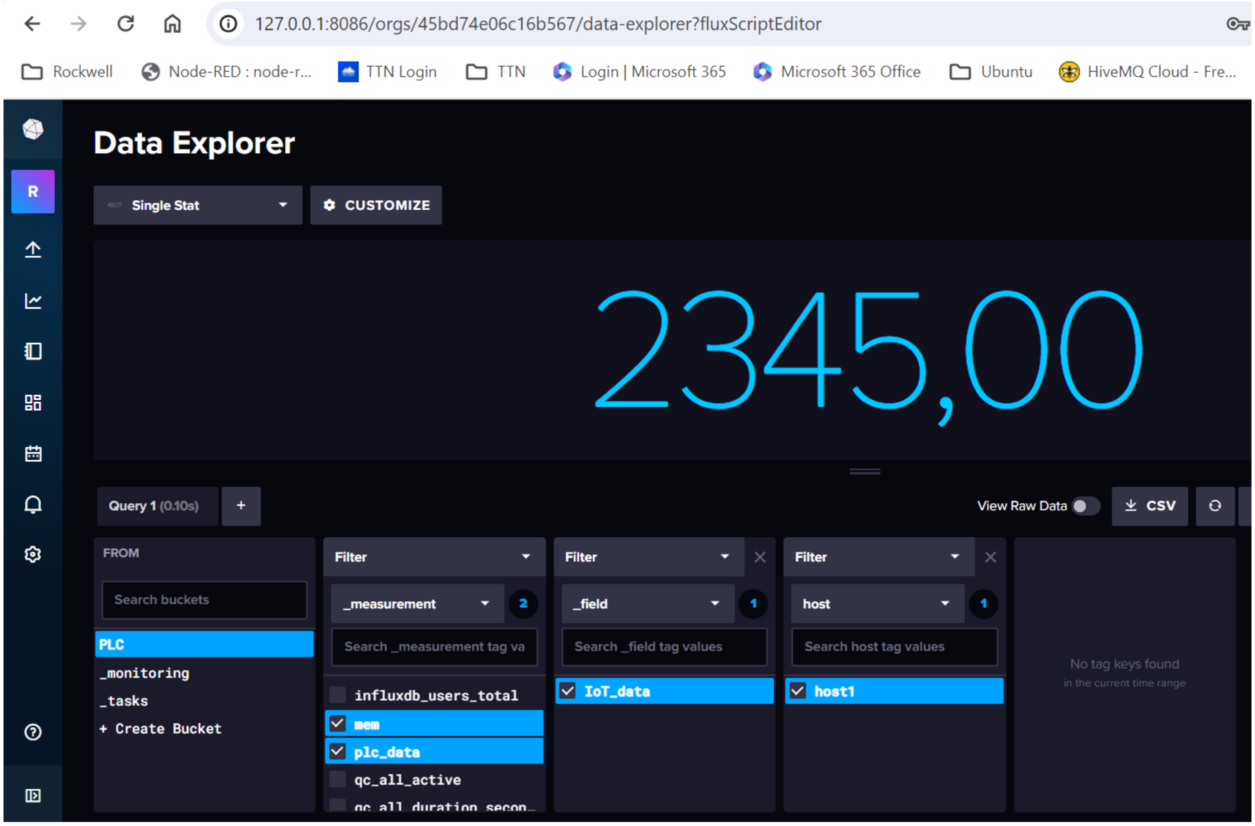
- □ influxdb_users_total
- ☑ mem
- ☑ plc_data
- □ qc_all_active
- □ qc_all_duration_secon…

Filter

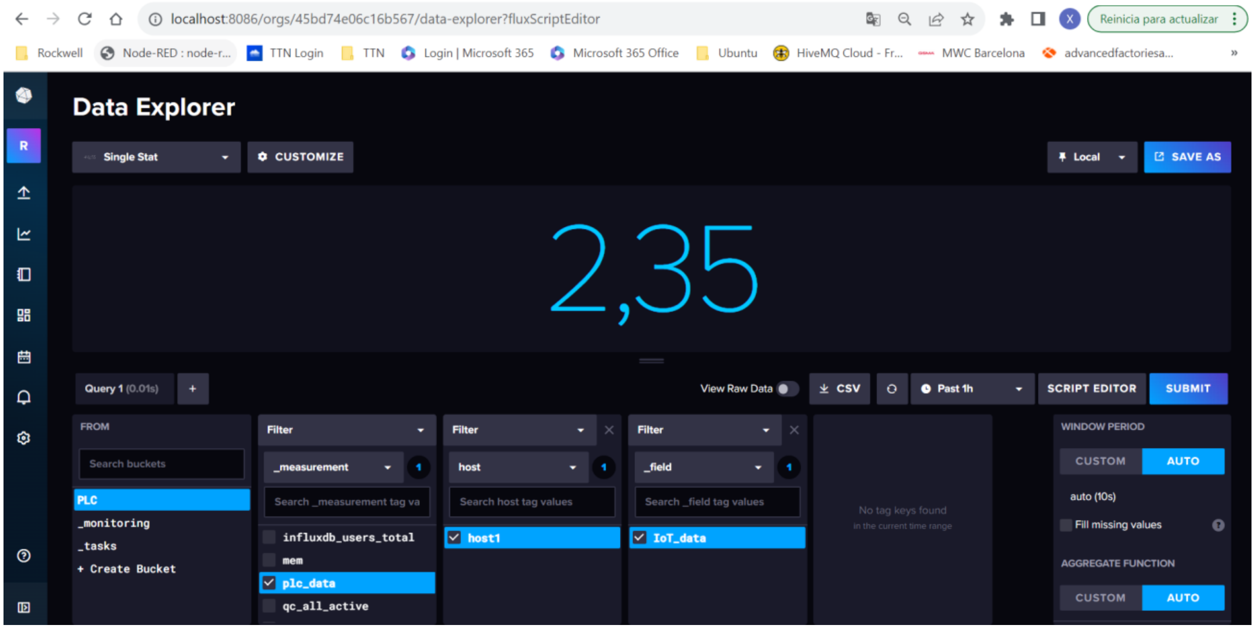
_field

Search _field tag values

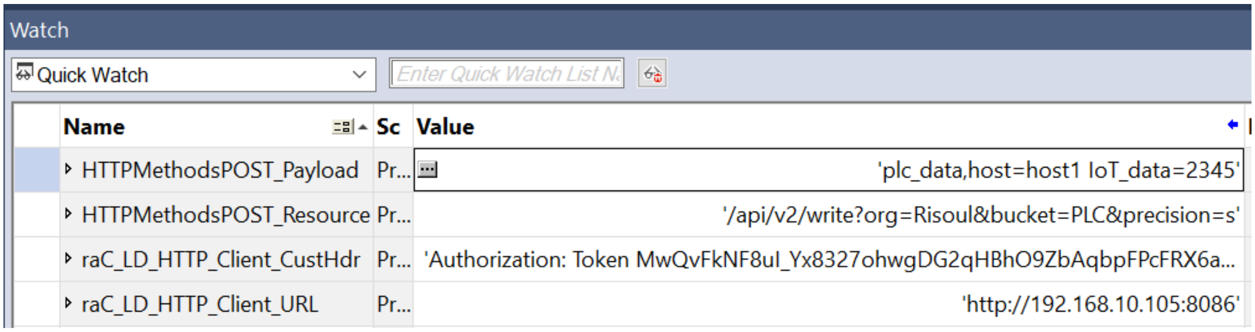
- □ IoT_data



Be careful with the comma



So these are the configuration



Name	Sc	Value
HTTPMethodsPOST_Payload	Pr...	'plc_data,host=host1 IoT_data=2345'
HTTPMethodsPOST_Resource	Pr...	'/api/v2/write?org=Risoul&bucket=PLC&precision=s'
raC_LD_HTTP_Client_CustHdr	Pr...	'Authorization: Token MwQvFkNF8uI_Yx8327ohwgDG2qHBhO9ZbAqbpFPcFRX6a...
raC_LD_HTTP_Client_URL	Pr...	'http://192.168.10.105:8086'

You can see the final result on this video

https://youtu.be/tDezJRUEjq4

A cloud database cannot be used since This HTTP does not support secure https calls