RemLab

Co-funded by the
Erasmus+ Programme
of the European Union

REMOTIFICATION OF LAB EXERCISES

Cooperative Remote Labs Virtual Campus for
Higher Education in Industrial Engineering

RemLab
Project
Guidelines

Introduction

Nowadays remote access to test rigs can be implemented in different ways, starting from a remote desktop approach and up to complex cloud-based solutions. All the approaches have quite different levels of complexity in development, support and maintenance, including various security and safety considerations. While the remote desktop approach might seem like a simple and viable option, usually this is completely unacceptable for organizations with local networks and domain policies, like universities. Due to various network and internet security issues and safety considerations for the hardware, it is lineally impossible for teachers to give full or partial access to the local computers and thus network even to their students, not to mention students from partner universities or outside students. Another limitation in providing remote access is the necessity of web development expertise, which is rarely the case for teachers and university staff, not dealing directly with that field of knowledge. In the case of subcontracting this part of the development process, there is a high risk of inadequate or delayed support and maintenance issues of the developed remote lab, which is also not desired or accepted by teachers and the university. Therefore, a different approach should be considered in the university setting, to provide remote access to lab rigs.

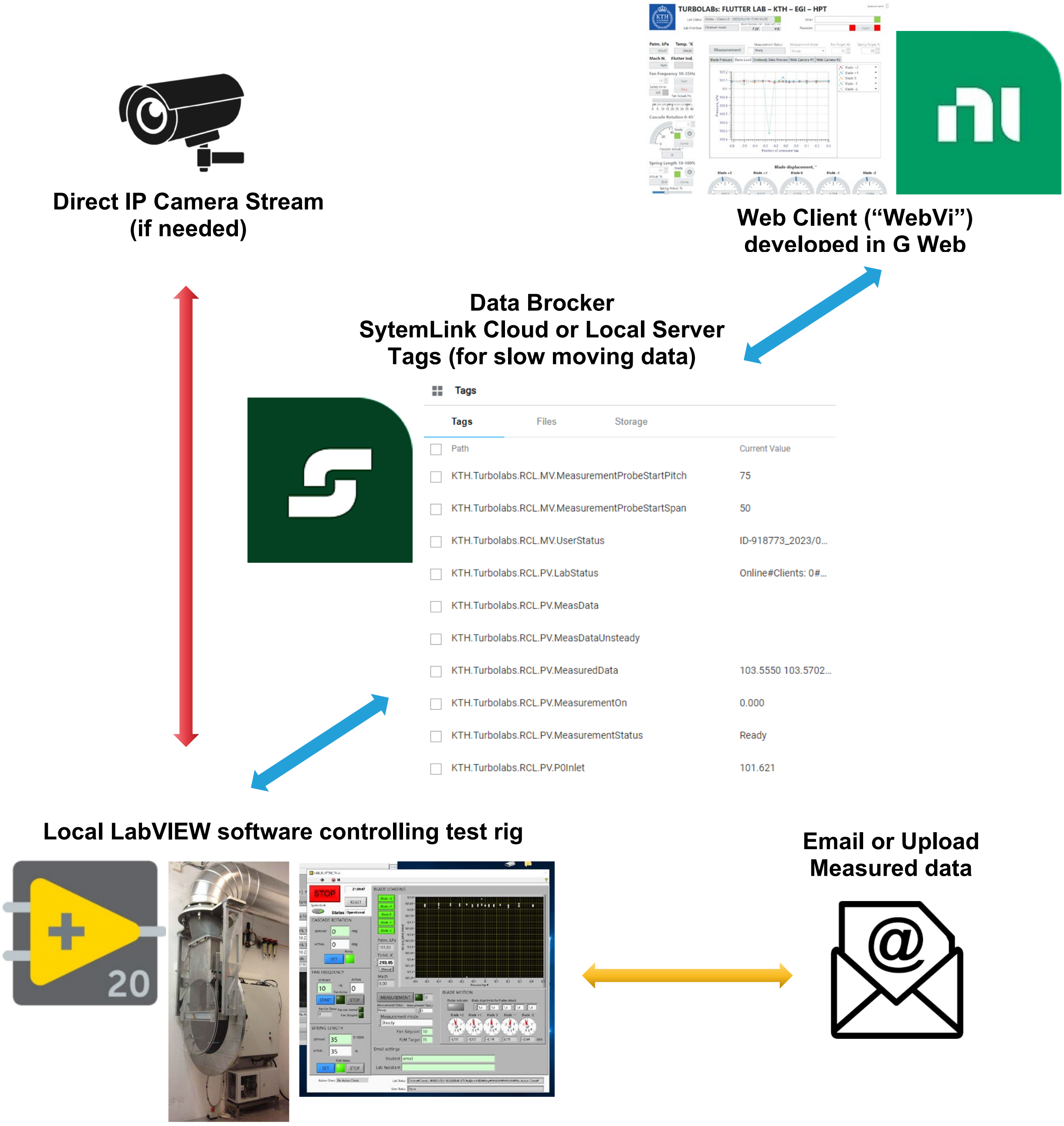
One of the desirable approaches would be using programming languages and environments familiar to teaching and university staff with the possibility of developing a solution that could provide remote access to test rigs, meeting all safety and security requirements. This would still involve a proper development phase, but the teacher would not be required to learn new programming languages or web programming or would not need to involve external professionals. Moreover, some of the required functionalities or modules could be simplified and adapted for a particular task eventually bringing it from lots of programming to less programming and more of configuration. It should be emphasized that it is very difficult to come up with some sort of universal and do-it-all solution for remote labs, and typically remote lab exercises are developed differently at different universities, depending on available in-house expertise (or lack of it). Within the RemLab project, we tried to work on and offer a potential solution for remote lab development. The proposed approach worked well for the project consortium and could be also implemented by other universities without all the hassle of the initial development phase we have already completed. The proposed solution allows teachers to develop remote interfaces for existing or new lab exercises and also set up and deploy all relevant data communication, webpage hosting, bidirectional data exchange and user management.

“Remotification” approach

While working on “remotification” of the lab exercises we have discovered a potential solution in the form of National Instruments G Web Development Software (formerly LabVIEW NXG Web Module) for creating web pages within the NI LabVIEW-like environment. The unique advantage of using G Web is that fully functional and modern-looking web pages with sufficient automation control and visualization capabilities could be developed with little to no knowledge of web programming, https://www.webvi.io/. The development process is very similar to local LabVIEW VI (virtual instruments) interface developments and thus will be familiar to teachers with former LabVIEW experience. We also believe that teachers without former LabVIEW experience will be able to use G Web Development Software after a few training sessions. To simplify adoption even further, we have created a template of a remote lab exercise interface, that could be used for learning or as a basis for developing new lab exercises.

Communication between the web interface and the test rigs could be also implemented in different ways due abundance of various communication protocols and solutions nowadays. But to keep the lab development process within the same programming environment we have considered various possibilities NI LabVIEW and related software packages could offer. Several options were considered and after various tests, we have settled on the NI Systemlink Tags solution, which works similarly to IoT MQQT protocol approach for data communication over the network.

The schematics of “remotification” approach is presented in Figure 1. The lab rig is controlled on local PC by LabVIEW with integrated Systemlink communication module or any other control software that could exchange data (process and manipulate variables[1]) with resident LabVIEW software with integrated Systemlink communication module. The lab data (process and manipulate variables) is being constantly exchanged with the data broker SytemLink Cloud using Tags. Tags in SystemLink transmit and store slow-moving measurement data. Typical use of tags is for tracking measurements, monitor system health, create alarms, and visualize data on dashboards. Lab client (WebVI) constantly exchanges data with data broker SytemLink Cloud, allowing to read the process variables and send the manipulate variable as use input.



Tags

Tags Files Storage

Path	Current Value
KTH.Turbolabs.RCL.MV.MeasurementProbeStartPitch	75
KTH.Turbolabs.RCL.MV.MeasurementProbeStartSpan	50
KTH.Turbolabs.RCL.MV.UserStatus	ID-918773_2023/0...
KTH.Turbolabs.RCL.PV.LabStatus	Online#Clients: 0#...
KTH.Turbolabs.RCL.PV.MeasData	
KTH.Turbolabs.RCL.PV.MeasDataUnsteady	
KTH.Turbolabs.RCL.PV.MeasuredData	103.5550 103.5702...
KTH.Turbolabs.RCL.PV.MeasurementOn	0.000
KTH.Turbolabs.RCL.PV.MeasurementStatus	Ready
KTH.Turbolabs.RCL.PV.P0Inlet	101.621

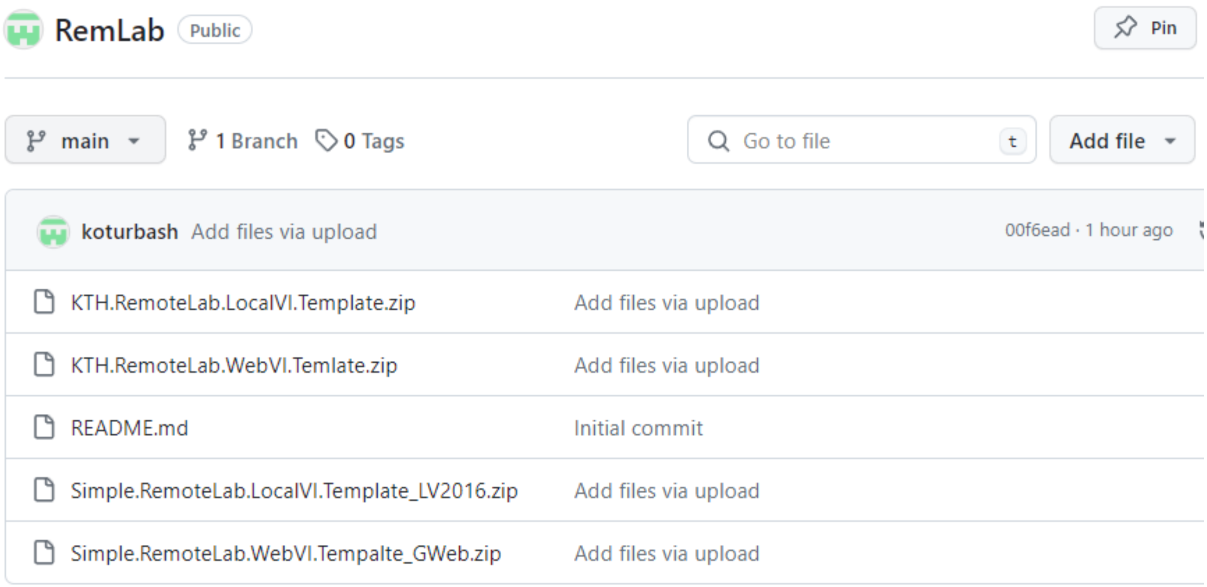
Figure 1 Schematics of “remotification” approach

[1] A process variable is the current measured value of a process or particular part of a process which is being monitored or controlled. Manipulated variable is the input that a process operator introduces to control or manipulate the process.

A typical delay of data exchange from lab rig to lab client in one direction is in the range of 1 sec. This delay is unavoidable and should be considered in the design of the lab exercise. Thus, fast changing data should not be transferred by tags without additional considerations. If fast changing data is relevant to lab implementation, then this data should be emailed to the client or uploaded into file sharing service. This could be accomplished by using an emailing module. Liver camera view could be arranged via tags also, but in this case the resolution and refresh rate is limited. In case live streaming is relevant to lab implementation, typical IP camera streaming should be used. The IP address of the camera and access credential could be transferred through tag to lab client.

How to use “remotification” tool

Developed “remotification” tool comprises of samples code or templates of lab exercises with all developed modules for communication, data exchange and user management. Currently there are two templates available: simple data exchange template and full fledged lab exercise exercise. All the template are available on Github and could be accessed through www.remlab.site.



koturbash Add files via upload 00f6ead · 1 hour ago

KTH.RemoteLab.LocalVI.Template.zip	Add files via upload
KTH.RemoteLab.WebVI.Temlate.zip	Add files via upload
README.md	Initial commit
Simple.RemoteLab.LocalVI.Template_LV2016.zip	Add files via upload
Simple.RemoteLab.WebVI.Tempalte_GWeb.zip	Add files via upload

Each template consists of two parts: local LabVIEW software (developed in LabVIEW 2019) and Web Client software (G-Web Development Software).

Open template and follow inbuilt instruction and comment to learn the principles of operation and develop you own “remotification” of lab exercises.

How to setup and run Web Client.

Two approaches to run web client and data brocket a available: SystemLink Cloud and Local SystemLink Server

The local LabVIEW software should always run on any Windows machine and requires internet access to the Internet, to exchange data with SystemLink Cloud. In the case of SystemLink Cloud usage, only local LabVIEW software should run on lab rig PC. SystemLink Cloud will host and run Tags and Web Client (WebVi). If local hosting of Web Client and local SystemLink Server is considered, the NI Web Server should be installed and running on same lab rig PC or another dedicated server PC. Please note that dedicated server PC is not required if no high user load is expected (1000+ concurrent users), which is not the typical usage case for lab exercises anyways.

1. SystemLink Cloud

SystemLink Cloud is a cloud-based data exchange platform offered by NI. It enables you to securely access, monitor, and interact with applications from anywhere in the world without the burden of hosting and managing a server yourself. Access SystemLink Cloud is arranged with your NI User Account credentials and all web application could be published for public access on the Internet.

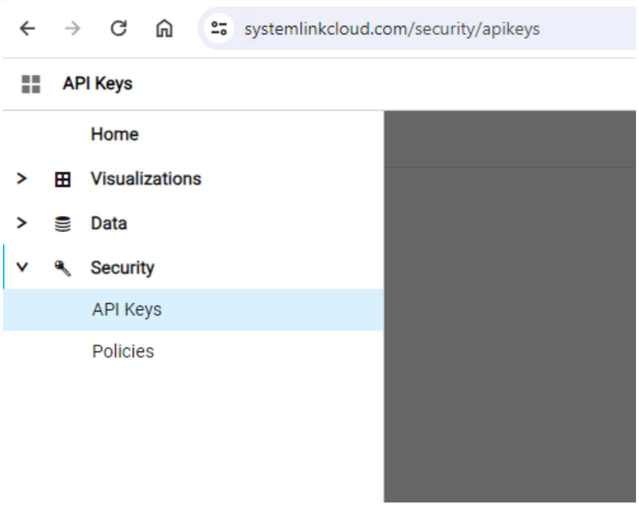
SystemLink Cloud could be accessed if you purchased G Web Development Software, LabVIEW NXG Web Module or SystemLink Server; or if you have an Enterprise Agreement or an Academic Site License, contact Sales to verify if you can access SystemLink Cloud.

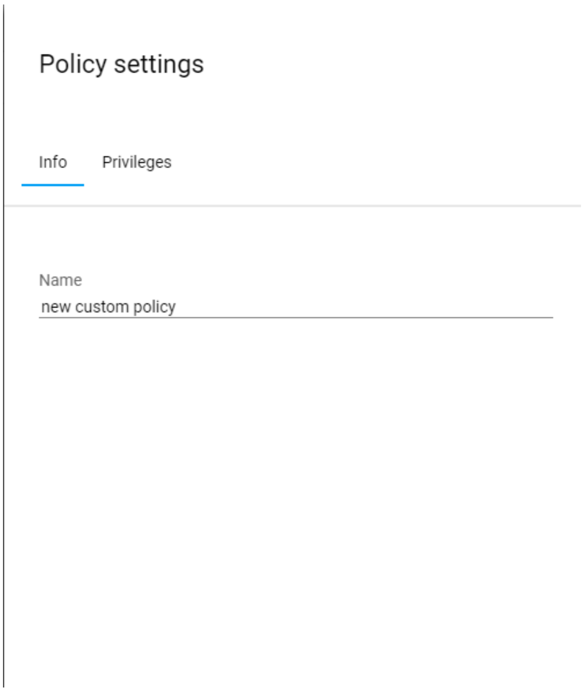
To build applications with APIs that communicate with SystemLink Cloud, you need to install one of the following product combinations with NI Package Manager: G Web Development Software or LabVIEW 2018-2021 with SystemLink Client 2021 R2. After installing the required products, launch LabVIEW, G Web Development Software or LabVIEW NXG Web Module and locate the SystemLink APIs on the Data Communication palette.

Connecting to SystemLink Cloud

As soon as you obtain a SystemLink Cloud serial number, you’re ready to access SystemLink Cloud. In order to connect your web application to access, monitor, and interact with data from a web browser following procedure should be completed:

- Create a new API key to authenticate your web application.
- Create a policy to set the permissions of a web application for accessing Tag.
- Assign a policy to an API key to define the actions a web application can make.
- Connect to SystemLink Cloud from a web application using LabVIEW, G Web Development Software or LabVIEW NXG Web Module.

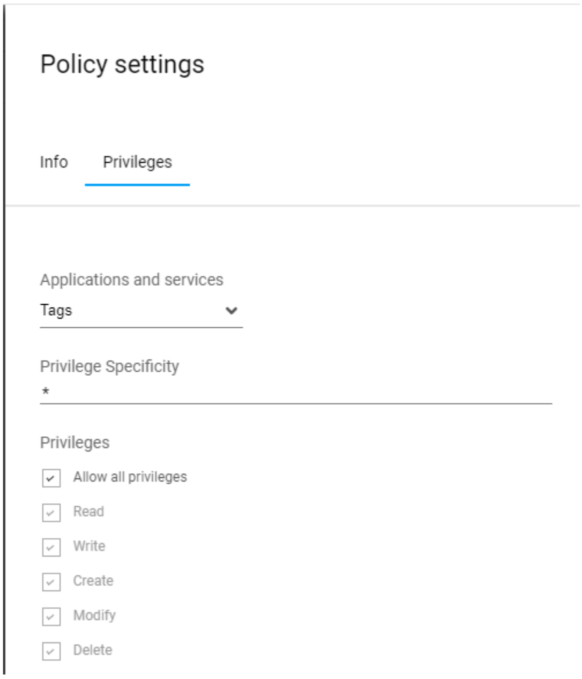




Policy settings

Info Privileges

Name
new custom policy



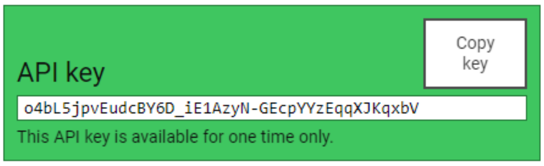
Policy settings

Info Privileges

Applications and services
Tags

Privilege Specificity
*

Privileges
☑ Allow all privileges
☑ Read
☑ Write
☑ Create
☑ Modify
☑ Delete

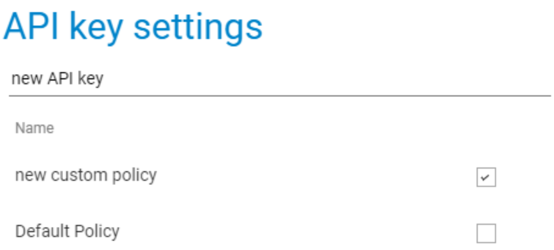


API key

Copy key

o4bL5jpvEudcBY6D_iE1AzyN-GEcpYYzEqqXJKqxbV

This API key is available for one time only.



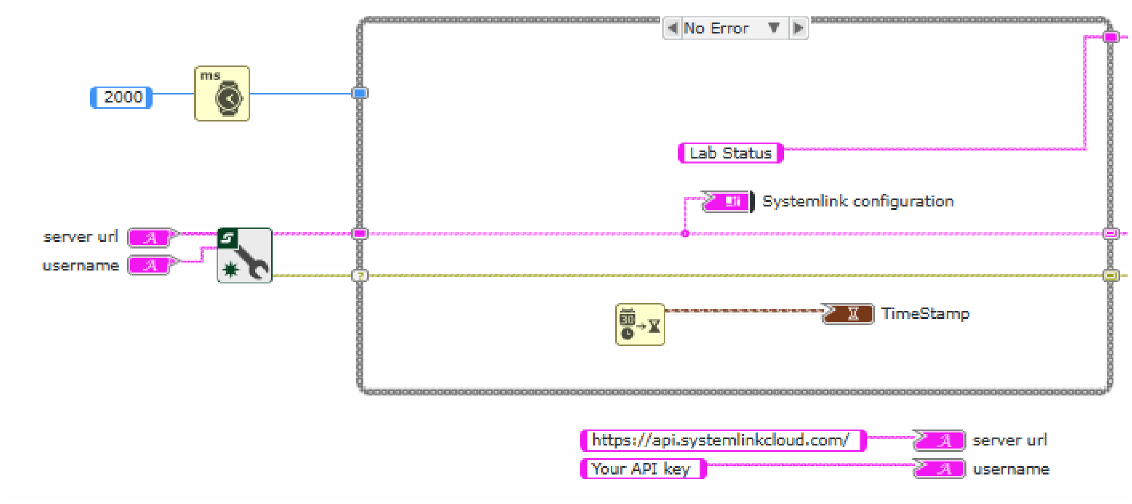
API key settings

new API key

Name

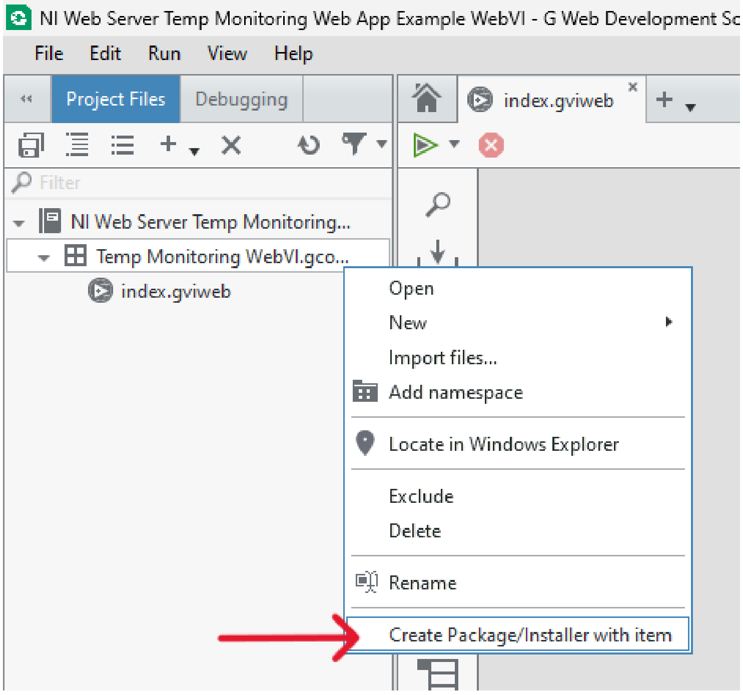
new custom policy ☑

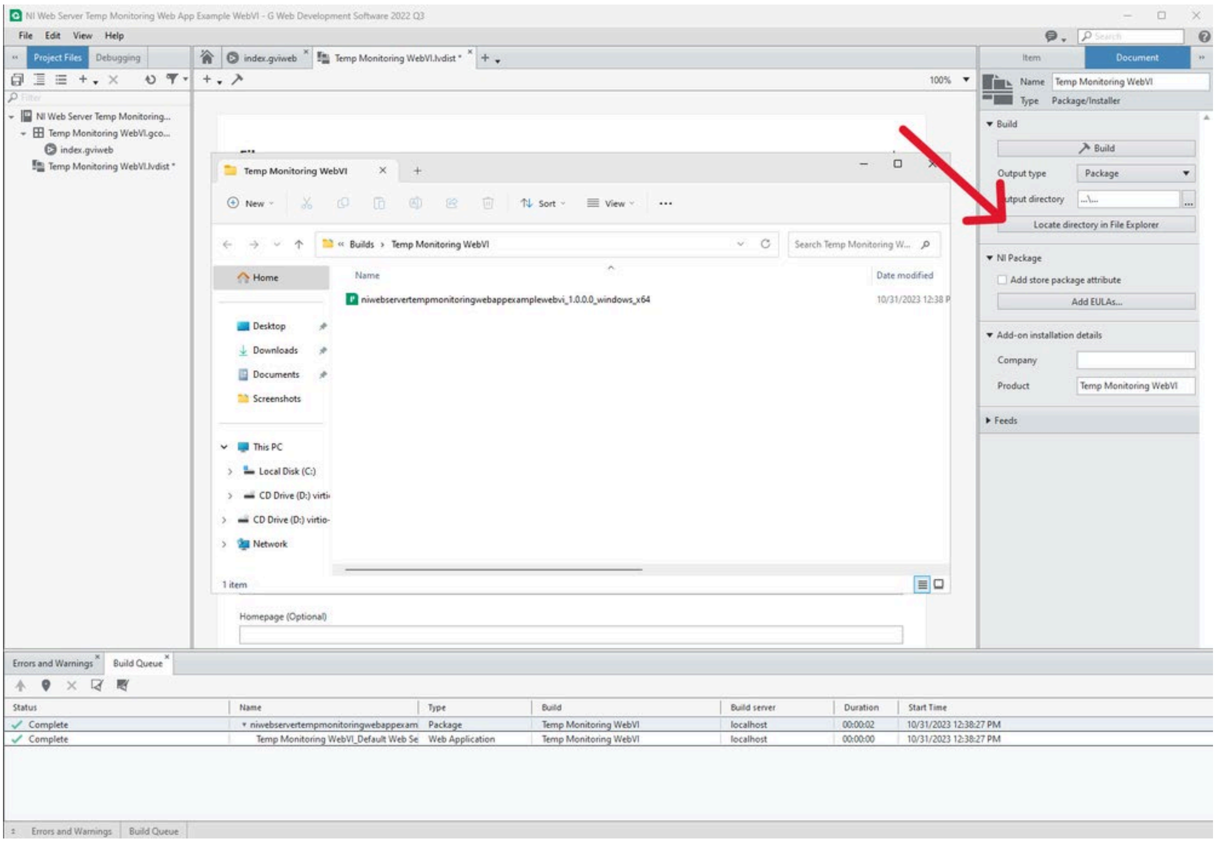
Default Policy □



Hosting a Web Application on SystemLink Cloud

For SystemLink Cloud to host your WebVi, you must build a package file (.nipkg) of your application and upload the package. Refer to Creating a Package in G Web Development Software or Creating a Package in LabVIEW NXG Web Module for more information.


NI Web Server Temp Monitoring Web App Example WebVI - G Web Development Sc
File
Edit
Run
View
Help
Project Files
Debugging
index.gviweb
Filter
NI Web Server Temp Monitoring...
Temp Monitoring WebVI.gco...
index.gviweb
Open
New
Import files...
Add namespace
Locate in Windows Explorer
Exclude
Delete
Rename
Create Package/Installer with item


NI Web Server Temp Monitoring Web App Example WebVI - G Web Development Software 2022 Q3
File
Edit
View
Help
Project Files
Debugging
index.gviweb
Temp Monitoring WebVI.lvdist *
NI Web Server Temp Monitoring...
Temp Monitoring WebVI.gco...
index.gviweb
Temp Monitoring WebVI.lvdist *
100%
Temp Monitoring WebVI
New
Sort
View
Builds › Temp Monitoring WebVI
Search Temp Monitoring W...
Home
Desktop
Downloads
Documents
Screenshots
This PC
Local Disk (C:)
CD Drive (D:) virti
CD Drive (D:) virtio-
Network
1 item
Item
Document
Search

Name	Date modified
niwebservertempmonitoringwebappexamplewebvi_1.0.0.0_windows_x64	10/31/2023 12:38 P

Homepage (Optional)

Name: Temp Monitoring WebVI
Type: Package/Installer

Build

Build

Output type: Package

utput directory: ...\...

Locate directory in File Explorer

NI Package

☐ Add store package attribute

Add EULAs...

Add-on installation details

Company:

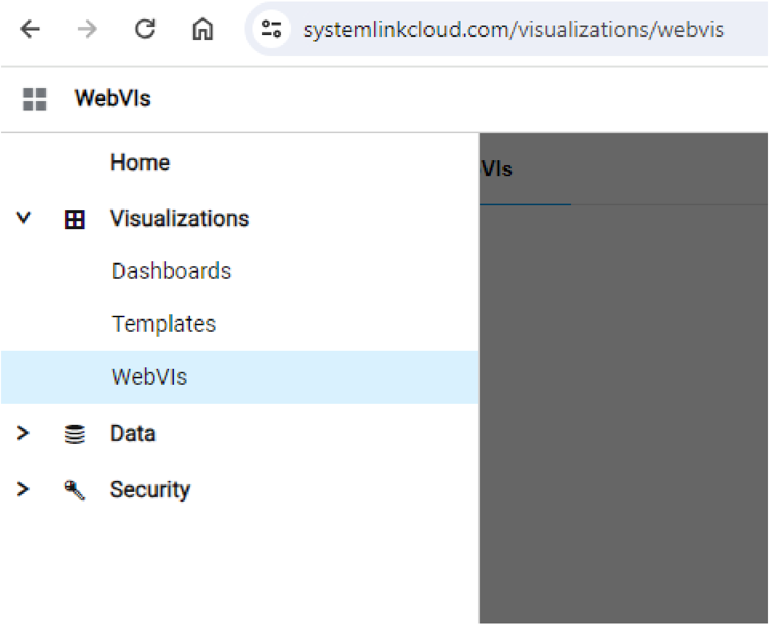
Product: Temp Monitoring WebVI

Feeds

Errors and Warnings | Build Queue

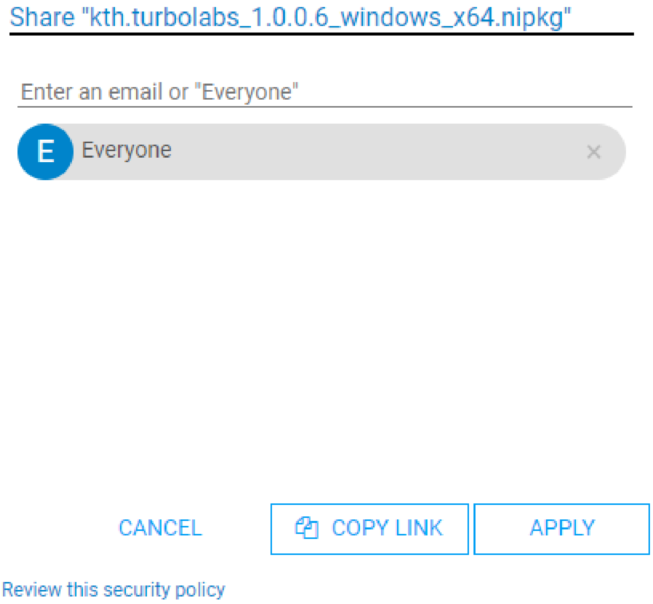
Status	Name	Type	Build	Build server	Duration	Start Time
✓ Complete	niwebservertempmonitoringwebappexam	Package	Temp Monitoring WebVI	localhost	00:00:02	10/31/2023 12:38:27 PM
✓ Complete	Temp Monitoring WebVI_Default Web Se	Web Application	Temp Monitoring WebVI	localhost	00:00:00	10/31/2023 12:38:27 PM

In order to upload your WebVi you can create new WebVi application or update existing one. Note, each web application shown in SystemLink WebVis has own unique web address.



systemlinkcloud.com/visualizations/webvis

Access your web application as web address for specific WebVI



Share "kth.turbolabs_1.0.0.6_windows_x64.nipkg"

Enter an email or "Everyone"

Everyone

2. Local SystemLink Server

Instead of running the data broker and hosting web client on NI SystemLink Cloud, a local web server could be arranged. This is especially relevant in no SystemLink Cloud license is available.

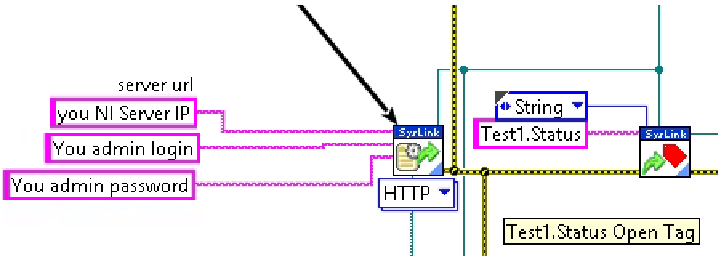
Required software: LabVIEW 2019 with LabVIEW SystemLink Toolkit (installs with G Web Development Software); G Web Development Software 2022 Q3 and SystemLink API (installs with G Web Development Software); NI Web Server (installs with G Web Development Software)

Setup of local server

There are several server configuration presets you can use for the NI Web Server. We recommend starting with “Simple local access” while developing your web application and then moving on to one of the other web accessible configurations, once you’ve verified your web application is working correctly.

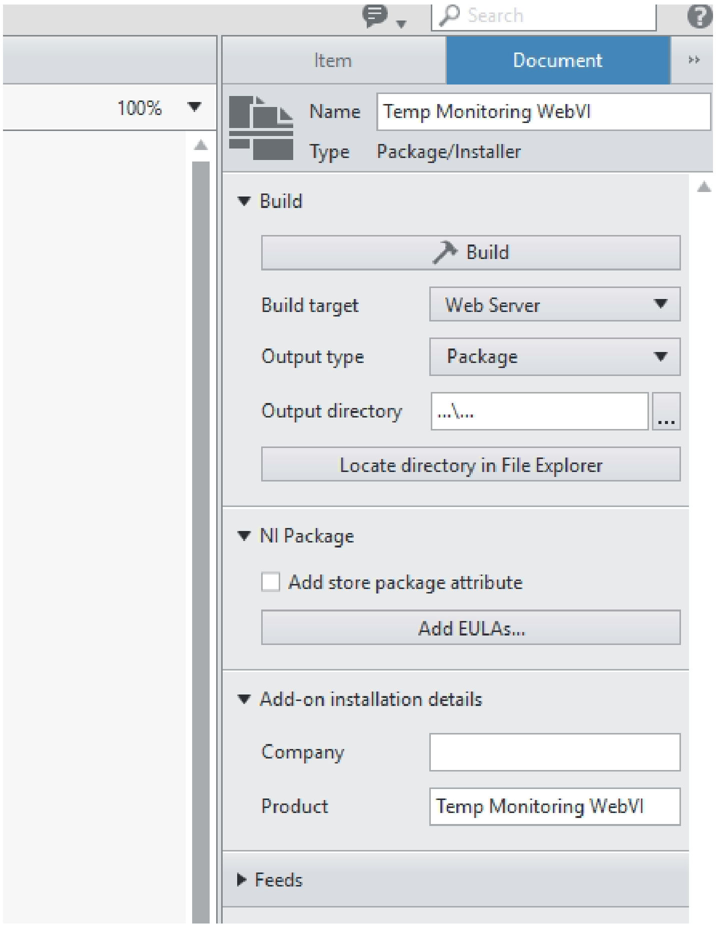
- Goto NI Web Server Guided Setup wizard
    - If opening the NI Web Server Configuration tool for the first time, it will automatically start you in the guided setup wizard.
    - If the NI Web Server has already been configured, go to the Summary tab then select Run guided setup to select preset settings to run the wizard again.
- Select Public Network Access and Press Next.
- Select which authentication providers you would like to use. Use this page in the NI Web Server manual to help you choose. The quickest way to get started is to only setup an admin account. Once selected, press Next.
- Select which HTTP port to use. The default port is 80, but can be changed based on your needs. Once selected, press Next.
- Configuration is complete. Press Finish to apply the configuration.

Link local LabVIEW Application to the created NI Web Server



Build and upload you WebVi to local NI Web server

Build your WebVi as described in previous steps and locate your WebVi as a webpage files (not package).

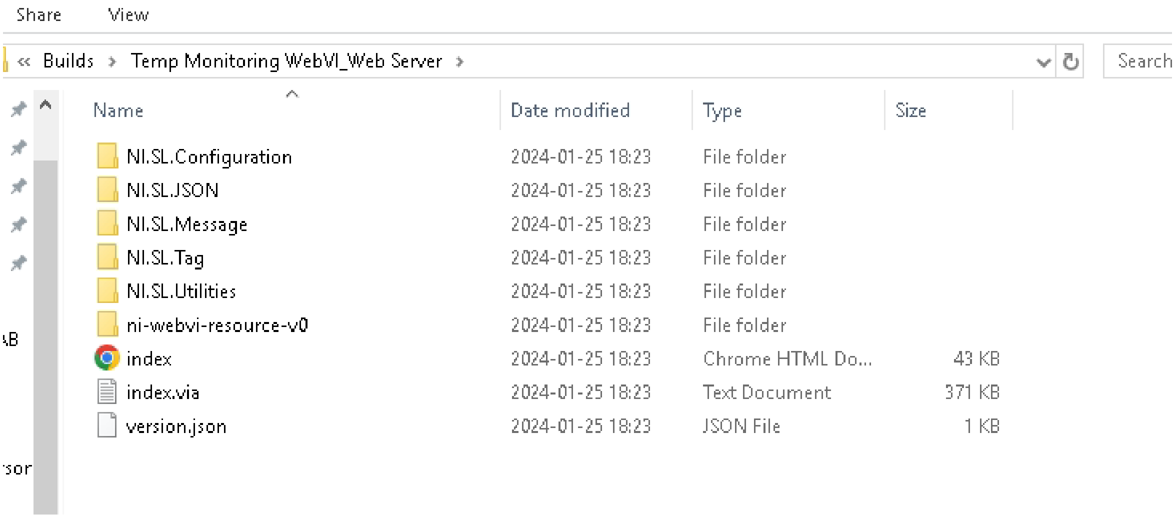

Search
Item
Document
100%
Build
Build
Locate directory in File Explorer
NI Package
Add EULAs...
Add-on installation details
Feeds

Name: Temp Monitoring WebVI
Type: Package/Installer

Build target: Web Server
Output type: Package
Output directory: ...\...

☐ Add store package attribute

Company:
Product: Temp Monitoring WebVI



« Builds › Temp Monitoring WebVI_Web Server ›

Name	Date modified	Type	Size
NI.SL.Configuration	2024-01-25 18:23	File folder	
NI.SL.JSON	2024-01-25 18:23	File folder	
NI.SL.Message	2024-01-25 18:23	File folder	
NI.SL.Tag	2024-01-25 18:23	File folder	
NI.SL.Utilities	2024-01-25 18:23	File folder	
ni-webvi-resource-v0	2024-01-25 18:23	File folder	
index	2024-01-25 18:23	Chrome HTML Do...	43 KB
index.via	2024-01-25 18:23	Text Document	371 KB
version.json	2024-01-25 18:23	JSON File	1 KB

Upload your built WebVi as a webpage files (not package) to NI Web Server at "C:\Program Files\National Instruments\Shared\Web Server\htdocs\Your web application name"

Access your web application as web address: http:\*Your Ni Web Server IP/ Your web application name*