Hands-On: Create Web-Ready User Interfaces With the LabVIEW NXG Web Module

Contents

Software Requirements **3**
Part A: Overview of System Components **3**
LabVIEW NXG 3
Parts of a Web Application 4
Part B: Introduction to WebVIs **5**
Create a WebVI From a Project Template 5
Explore the Web Application Project 6
Create a WebVI 7
Upload WebVI to Cloud Host 13
Part C: Using Flexible Layout **16**
Introduction to Flexible Layout 16
Change WebVI to Flexible Layout 16
Part D: NI Web Server **18**
Part E: Run the LabVIEW Application **19**
Run the LabVIEW 2019 Temperature Monitoring VI 19
Part F: Complete Web Application Code **20**
Overview of Example Code Files 20
Complete the Web Application Code 21
Part G: Run the Temperature Monitoring Web App **24**
Run and Explore the WebVI 24
Appendix A – Demo Data Server **26**
Appendix B – NI Web Server Configuration **26**
Configure the NI Web Server With the Guided Setup Utility 27
Apply and Restart the Web Server 28

Software Requirements

- LabVIEW NXG 3.1
- LabVIEW NXG Web Module 3.1
- LabVIEW 2019 (to connect a WebVI to an existing application)
    - Note: the SystemLink API supports LabVIEW 2015 and newer, but the code provided with this hands-on was developed in LabVIEW 2019.
- Internet connection

Part A: Overview of System Components

LabVIEW NXG

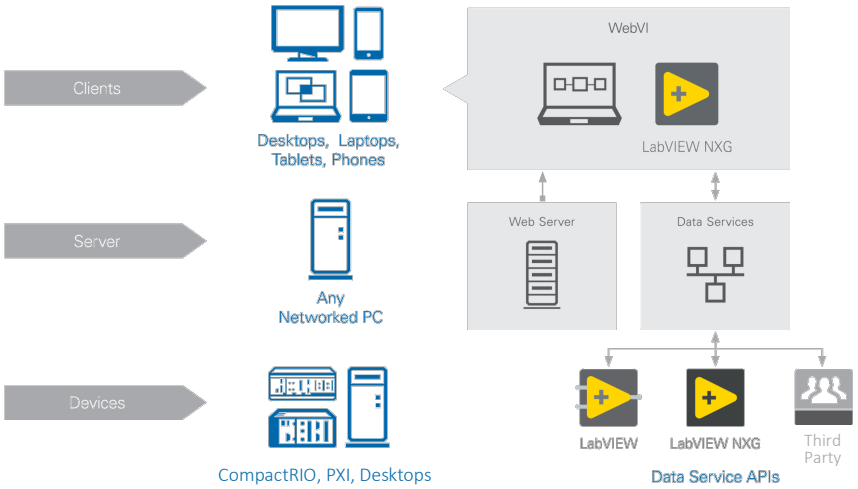
LabVIEW NXG Web Module

Expand LabVIEW systems engineering software with the LabVIEW NXG Web Module to create web applications that run in any modern web browser without plug-ins or installers. This module helps you quickly visualize data from distributed systems through drag-and-drop engineering user interface (UI) widgets, intuitive communication mechanisms, and secure hosting, all based on industry-standard technologies.

WebVI

Use WebVIs to create web applications that run natively within a web browser. Design UIs through drag-and-drop, high-performance engineering widgets that are built on standard web technologies (HTML, CSS, and JavaScript). With the openness of WebVIs, you can import content from across the Internet or export code to mainstream web development tools. You can program WebVIs in a similar manner to programming normal VIs, but with the added HTML source pane that allows custom HTML modifications, if necessary.

Parts of a Web Application



The diagram above shows the various software components running on the different types of hardware in a web application system.

Clients

- The user interacts directly with client devices, such as a computer or mobile devices running a web browser.
- WebVIs run on clients because they are compiled to execute natively within a web browser.
- In general, clients retrieve data from a server and display it to a user and/or send data to a server. It can also execute functions.

Server

- A server delivers WebVIs to the clients and provides information to WebVIs through a data service.
- These servers can run on any PC with network access that allows them to communicate with clients.
- If you want to avoid managing the server yourself, you can use services such as SystemLink Cloud from National Instruments.

Measurement Devices

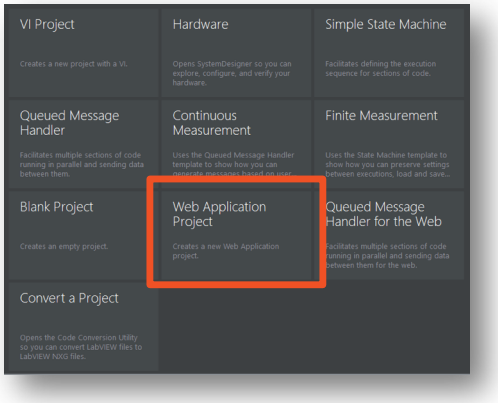
- Measurement devices communicate with WebVIs via data services, which can be implemented in LabVIEW, LabVIEW NXG, or third-party services.

Part B: Introduction to WebVIs

Create a WebVI From a Project Template

1. Launch LabVIEW NXG.

2. Click **Web Application Project**.
    a. If the Web Application Project template isn't visible, verify that you're on the Projects (not Learning) page and that you have the NXG Web Module installed.



3. Name the project "NXG Web App."

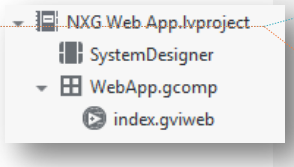
4. Select **Create**.

5. Save your project.

Explore the Web Application Project

SystemDesigner

SystemDesigner primarily provides a graphical representation of your hardware system and enables intuitive configuration, software organization, deployment, and documentation.



Commented [CZ1]: Is this index in 3.1? I am testing this in 3.0 and the title is Main.gviweb

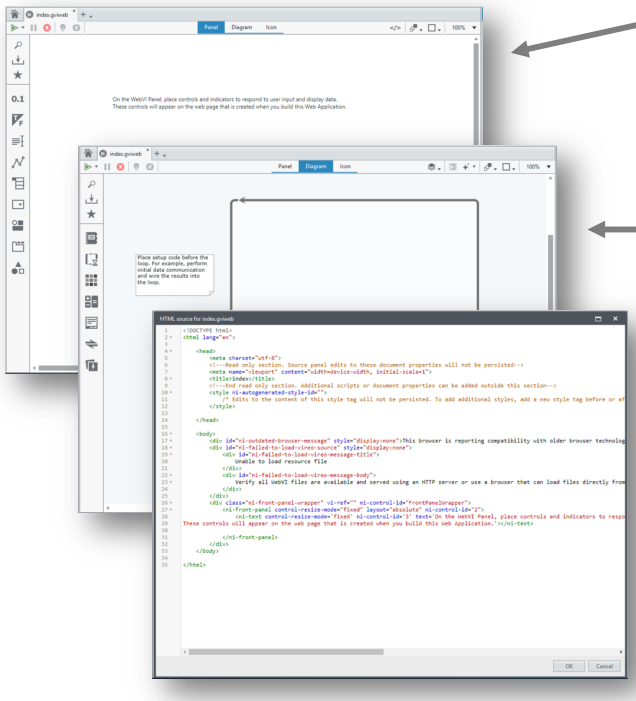
Commented [CR2R1]: Yes, it changed in 3.1.

Application document (.gcomp)

Use the WebApp.gcomp document to define and build your web application. The application document also serves as the container for all the files in your application.

WebVI (.gviweb)

The WebVI is a specialized VI that generates HTML, JavaScript, and CSS files when you build a web application. Each WebVI marked as top level in your application generates one HTML file, which corresponds to one web page after you build your web application.



Panel

The **Panel** window is the user interface for the WebVI. This is what users see when they access the web application in a browser.

Diagram

The **Diagram** view contains the code that executes when running a WebVI.

HTML

The **HTML** view </> contains the HTML code used by the WebVI in the built web application. The HTML code is generated automatically as you author the WebVI panel and diagram, but you can edit certain sections to gain custom functionality.

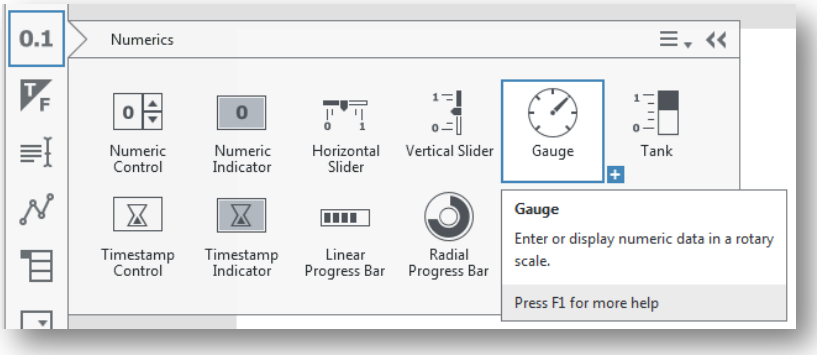
Create a WebVI

Create a User Interface panel

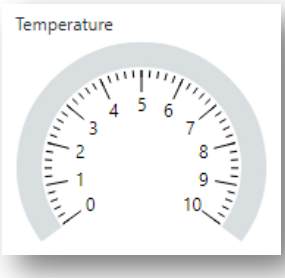
Complete the following steps to create a panel that can display a temperature value.

6. Delete the template text on the panel of *index.gviweb*.

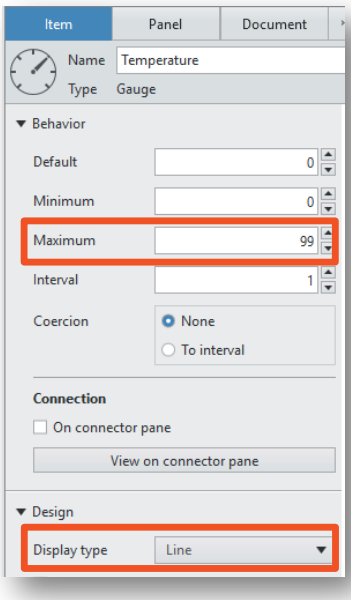
7. Select the **Gauge** in the *Numerics* palette and place it onto the panel.



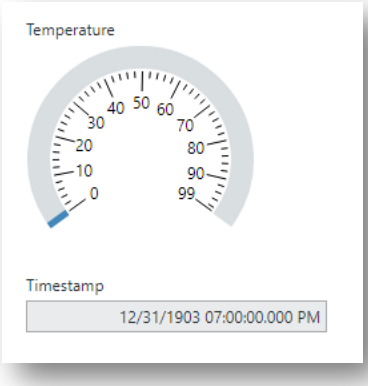
8. Name the Gauge *Temperature*.



9. In the *Item* configuration pane on the right side of the editor, go to the *Behavior* section and set the *Maximum* to 99. Go the *Design* section and change the *Display type* to Line.



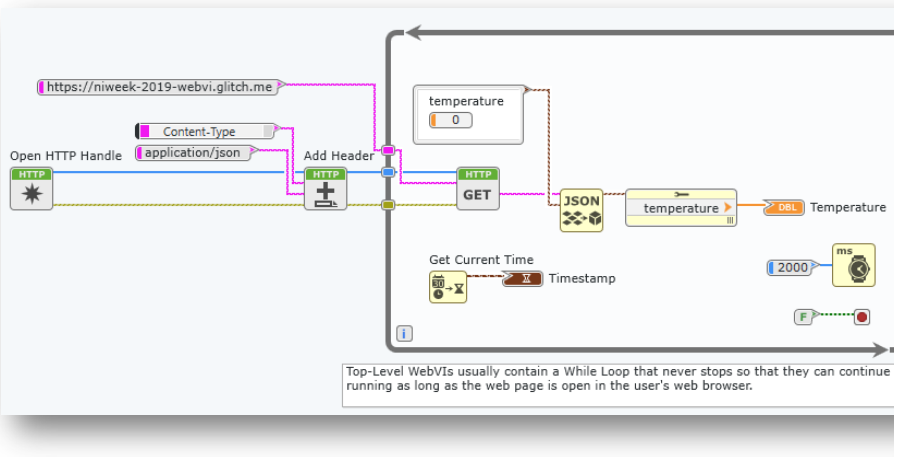
10. Select the **Timestamp Indicator** in the *Numerics* palette and place it onto the panel. Your panel should look like this:



Create a diagram

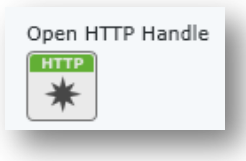
Complete the following steps to read a temperature from a data server and display it. For this exercise, National Instruments is publishing data that anyone can view. We will use HTTP, which is a standard method to retrieve information from a web server. Later exercises will show alternative methods and introduce access control.

Your completed diagram will look similar to this:

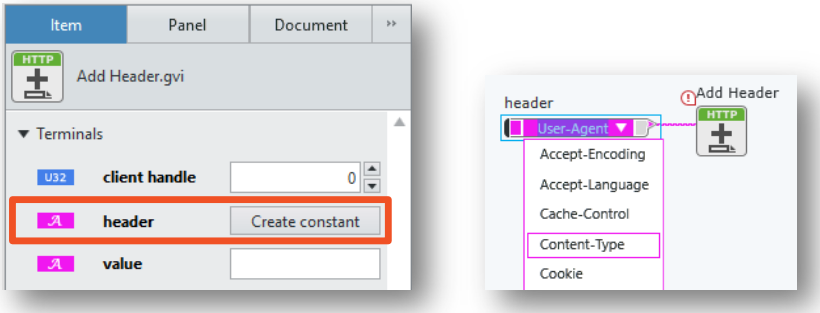


11. Navigate to the diagram by selecting the **Diagram** view or pressing <**Ctrl-E**> on the keyboard to toggle between the Panel and Diagram views.

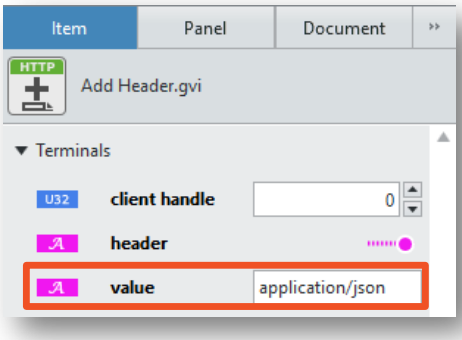
12. Place **Open HTTP Handle** from the *Data Communications»Internet»HTTP* palette on the diagram to the left of the While loop. Note: because we will be reading a public value, you do not need to wire the username or password inputs.



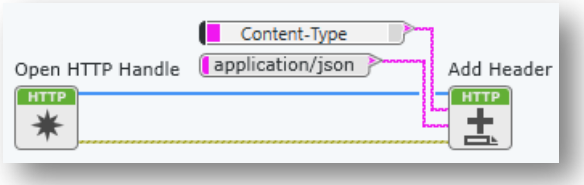
13. Place **Add Header** from the *Data Communications»Internet»HTTP»Utilities* palette on the diagram. In the *Item* configuration pane, press the *Create Constant* button next to the header to create and connect a constant on the diagram. Change the constant value to *Content-Type*.



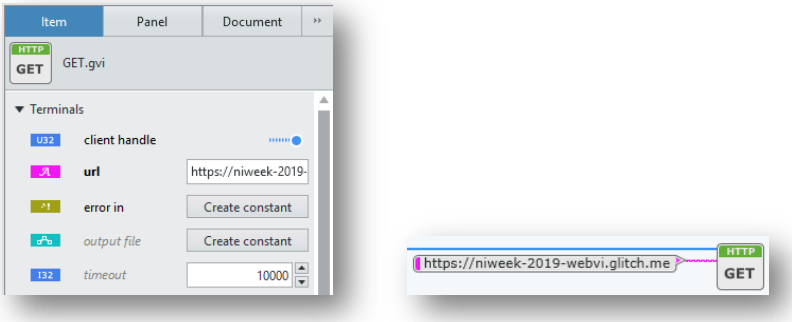
14. In the *Item* configuration pane, set the value to **application/json**. LabVIEW NXG will automatically create a constant on the diagram.



15. Connect the functions as showing in the image.



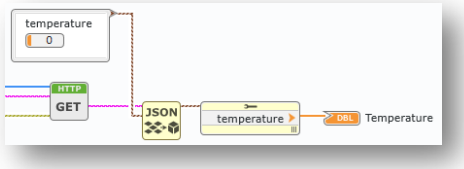
16. Place **GET** from the *Data Communications»Internet»HTTP* palette inside the While loop and connect the *client handle* from the output of Add Header. In the *Item* configuration pane on the right, set the *url* to **https://niweek-2019-webvi.glitch.me**.



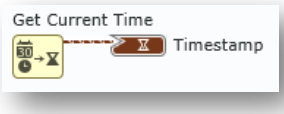
17. Place **Unflatten from JSON** from the *Data Types»String»Conversion* palette. Connect the *body* output of *GET* to the *JSON string* input. JSON (JavaScript Object Notation) is a standard format for exchanging data through HTTP.

18. Place a **Cluster Constant** from the *Data Types»Cluster* palette. Select a **Numeric Constant** from the *Data Types»Numeric* palette and place it inside the cluster constant. Name the numeric *temperature*. Connect it to the *type and defaults* input of *Unflatten from JSON.*

19. Place **Cluster Properties** from the *Data Types»Cluster* palette and the *Temperature* gauge terminal from the *Unplaced Items* palette. Wire the *value* from *Unflatten from JSON* to the *Cluster Properties* and then wire its output to the *Temperature* gauge terminal.

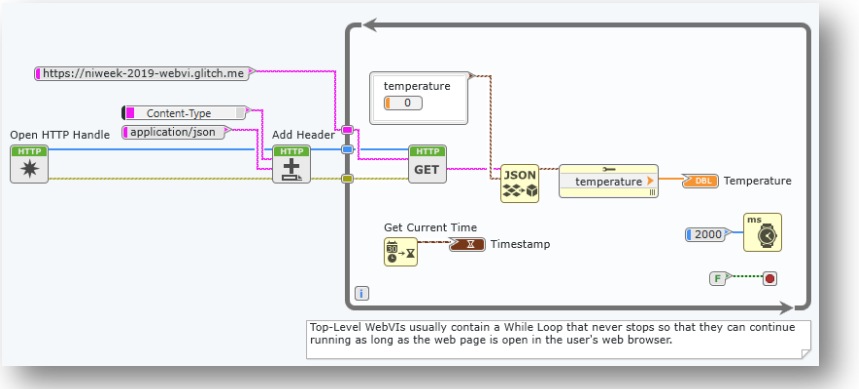


20. Place **Get Current Time** from the *Data Types»Timestamp* palette and the Timestamp terminal from the *Unplaced Items* palette. Connect them as shown in the image.

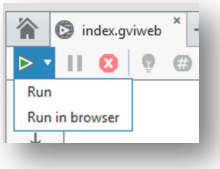


21. Select the **Wait** on the diagram. In the *Item* configuration pane on the right, set the *milliseconds to wait* value to 2000. LabVIEW NXG will automatically update the constant on the diagram.

22. Your diagram should now be complete.



23. To run the WebVI, you can choose to either **Run** the WebVI inside LabVIEW NXG, or you can choose to **Run in browser** which will build your WebVI and automatically open the local build in your default browser. Note that the LabVIEW NXG Web Module supports the following web browsers: Edge, Chrome, Firefox, and Safari.

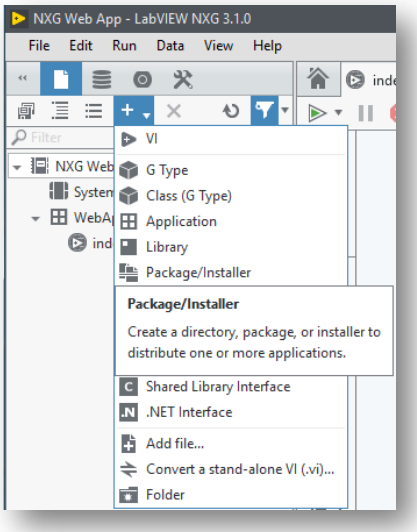


24. *Troubleshooting*: if the data server is unavailable then the web app will display a zero for the temperature. If this is happening because the demo data server at https://niweek-2019-webvi.glitch.me has exceeded its connection limit, refer to Appendix A for instructions on easily creating your own temporary server.

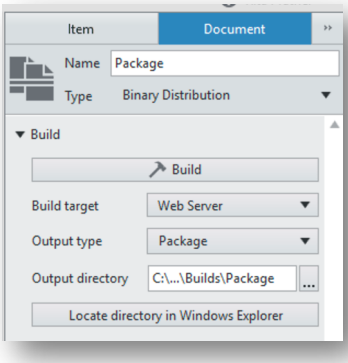
25. Stop your WebVI or close the web browser page if you ran it in a browser.

Upload WebVI to Cloud Host

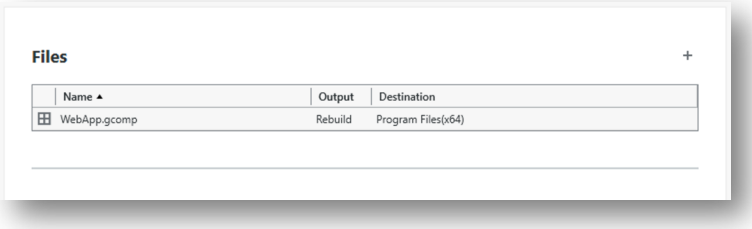
26. Let's host your WebVI in the cloud so that you can open it on your phone or other internet-connected device. For this step, we need to first build this web application into a Package. Add a Package to your project as shown below:



27. In the *Document* configuration pane on the right side of the editor, change the Build target to **Web Server** and the Output type to **Package**.



28. Add your WebApp.gcomp to the Files section of the Package document by clicking the + button.

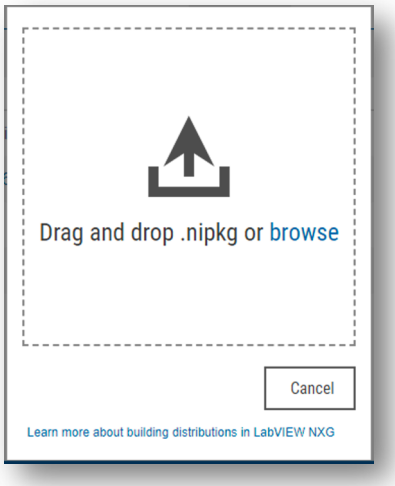


29. Build the Package by clicking the  button.

30. Once the Package has been built successfully, click the **Locate directory in Windows Explorer** button to see the output file.

31. Open a web browser and go to systemlinkcloud.com. Navigate to the Web apps section and log in. If you haven't signed up for SystemLink Cloud yet, start a free trial now. Note that if you already have a license for the LabVIEW NXG Web Module through your Alliance Partner Software suite or other license, you are entitled access to SystemLink Cloud with no extra charge, as long as you stay on SSP.

32. Click the **+ Upload** button on the top left of the window and either drag your built package onto the popup window or browse to the package manually.



33. Once you have successfully uploaded the package, you can view the built web application in your browser. You can also change the web application sharing settings from Private (default) to Public, or share the web application directly via email.

    If you have an internet connected device such as a phone or tablet with a web browser, you can view the web application hosted by SystemLink Cloud with either of these alternative methods:

    a. Navigate to systemlinkcloud.com on that device, log in, and go to your web apps. Select the app you want to view.
    b. Navigate directly to the application's URL on that device. You can bookmark and share this URL the same way you would with any website address.

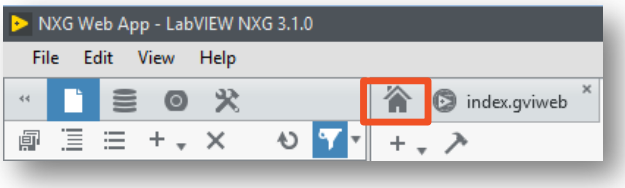
34. Save the project.
    Select **File»Save all** or press <**Ctrl-Shift-S**> on the keyboard.

Part C: Using Flexible Layout

When creating web applications for a variety of clients, it is important to consider how the UI will appear on different devices with different screen sizes and orientations. The flexible layout option for WebVI panels can help you build one panel that works well on different screens.

Introduction to Flexible Layout

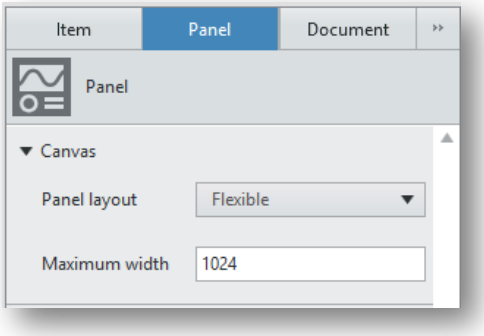
1. Click the Home button in the document tab bar to show the main landing page of LabVIEW NXG.



2. Go to the Learning tab. Then click on the Lessons tab and navigate to the Building Applications tile.
3. Open the workbook titled Designing a UI for Different Screen Sizes.
    a. Complete the workbook to become familiar with Flexible Layout.
    b. When iterating on the desired Flexible Layout behavior, don't forget to use the **Run in browser** option for the Run button; this is the quickest way to view how your WebVI changes in the browser when you resize it.

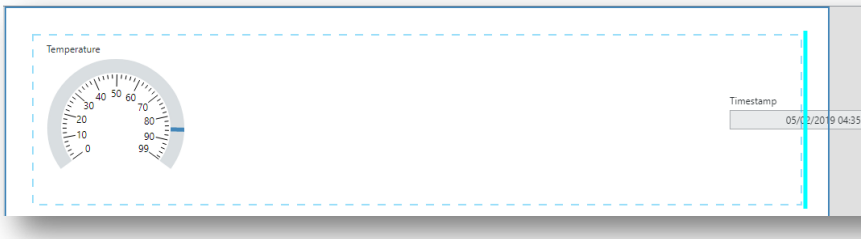
Change WebVI to Flexible Layout

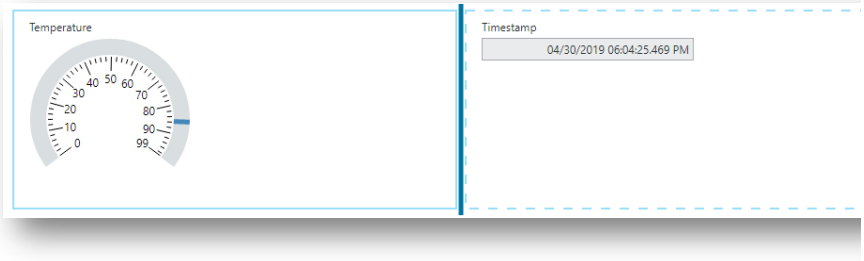
4. After you've completed the workbook, close the project and return to your NXG Web App project. Go to the panel of *index.gviweb*.
5. In the right-hand side of the editor, switch to the Panel tab of the configuration pane. Change the layout option from Absolute to Flexible.



6. LabVIEW NXG automatically moves your controls to *Unplaced Items* so that you can build your panel with your desired resize behavior.

7. Place **Temperature** and **Timestamp** from *Unplaced Items* such that they are in two side-by-side layout containers. You will need to place the Timestamp on the right edge of your panel in order to create the second layout container.





8. Run your WebVI in the browser and resize the window smaller to see the Timestamp automatically move beneath the Gauge. This simulates what the WebVI would look like on a small screen like a mobile device.

Part D: NI Web Server

In the next few sections, we will explore how to connect WebVIs to a desktop LabVIEW application.

Typically, a web application and desktop application communicate via shared storage on a server.

You can use any server you choose, but NI recommends using either SystemLink Cloud or the NI Web Server.

- **SystemLink Cloud** is an NI-hosted service in a secure, scalable cloud-computing environment that removes the need for additional hardware to maintain server infrastructure and manage web server configuration and security.
- You can set up the **NI Web Server** on any Windows computer. Use it when you want to manage your own server from within your network.

For this exercise, we'll use the NI Web Server, which is already running on the session computers. If you are using your own computer for these exercises, refer to Appendix B to setup the NI Web Server before proceeding.

The NI Web Server has been configured for *Simple Local Access* with an admin password of "test." For actual applications, NI recommends using a more secure password.

To learn more about the NI Web Server configuration options, please refer to the white paper *Security in NI Web Technology* which is available online at
http://www.ni.com/en-us/innovations/white-papers/18/security-in-ni-web-technology.html

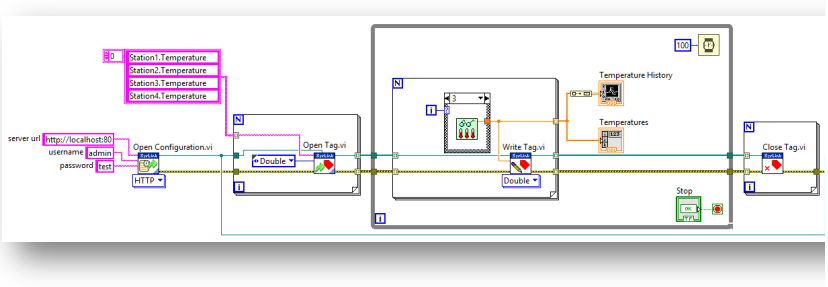
Part E: Run the LabVIEW Application

In this exercise, a LabVIEW 2019 application is monitoring the temperature of four stations. Since it is a desktop application, it could be connected to measurement devices, but the provided code simulates the temperature measurements. It publishes the temperature values to the NI Web Server.

Run the LabVIEW 2019 Temperature Monitoring VI

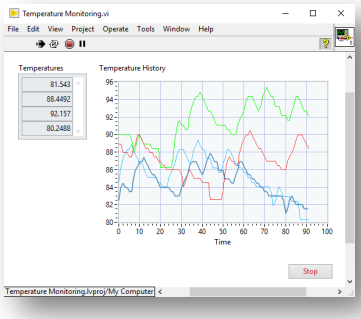
1. Navigate to the the Temperature Monitoring server code that is written in LabVIEW 2019 by going to Desktop\Web Module Hands-On\LabVIEW 2019 code. Open the **Temperature Monitoring** project.

2. Open **Temperature Monitoring.vi.** Open the VI diagram by selecting **Window»Show Block Diagram** or pressing <**Ctrl-E**> on the keyboard.



3. You do not need to edit this diagram, but you can review it to understand its functionality. In addition to simulating temperature measurements for four stations, this VI uses functions from the *Data Communication»SystemLink* palette to open communication with the NI Web Server and publish those simulated values to Tags. Note that the SystemLink API comes with the LabVIEW NXG Web Module; you do not have to purchase a SystemLink Server in order to use this API.

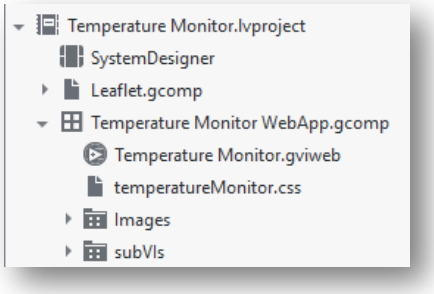
4. Select the **Run Arrow** or press <**Ctrl-R**> on the keyboard to run the code. View the front panel to ensure it is working. Leave this VI running while you complete the rest of the exercise.



Part F: Complete Web Application Code

Overview of Example Code Files

Open the partially completed example code by going to Desktop\Web Module Hands-On\LabVIEW NXG code. Double click the Project called Temperature Monitor.



Temperature Monitor WebApp.gcomp

The Temperature Monitor WebApp.gcomp is a component that groups the files of the web application together. It is used to build the web application.

Temperature Monitor.gviweb

The Temperature Monitor.gviweb is the web application's main UI. It displays temperature data retrieved from the server to the user of the client, in this case a browser.

temperatureMonitor.css

This CSS file includes visual styling that can be applied to a WebVI panel. The WebVI's HTML source references this file using a <link> tag.

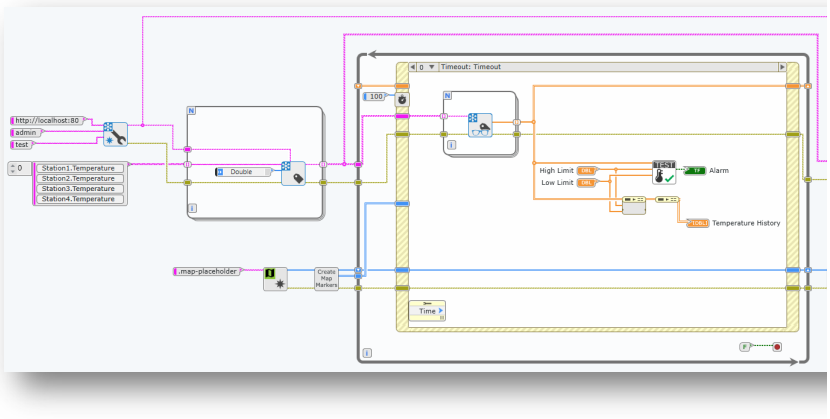
Leaflet.gcomp

This component provides an API to integrate content from Leaflet, an open-source JavaScript library for interactive maps. Learn more about Leaflet at leafletjs.com. Note that this component wraps the library in a G interface so that you can use it without any knowledge of the underlying JavaScript.

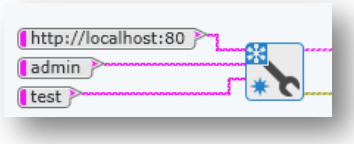
Complete the Web Application Code

1. Open Temperature Monitor.gviweb. Navigate to the diagram by selecting the **Diagram** view or pressing <**Ctrl-E**> on the keyboard to toggle between the Panel and Diagram views.

2. In the following steps, you will modify the diagram to match this image:

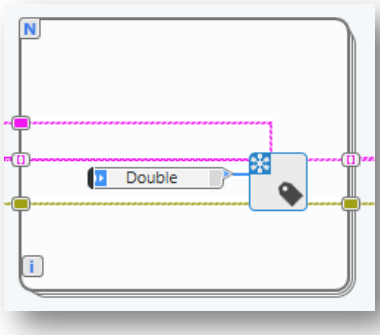


3. Place **Open Configuration** from the *Data Communications»SystemLink»Configuration* palette. Connect the wires as shown in the image. Note that the SystemLink API comes with the LabVIEW NXG Web Module; you do not have to purchase a SystemLink Server in order to use this API.

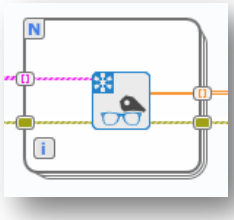


When prompted to choose between User and API Key, choose User. The API Key option is used when sending data to SystemLink Cloud rather than the NI Web Server.

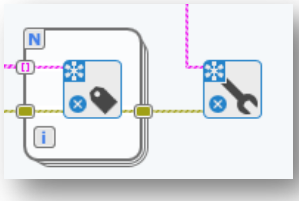
4. Place **Open Tag** from the *Data Communications»SystemLink»Tags* palette. Connect the wires as shown in the image.



5. Place **Read Tag** from the *Data Communications»SystemLink»Tags* palette. When prompted to choose the *Function configuration*, choose Double. Connect the wires as shown in the image.

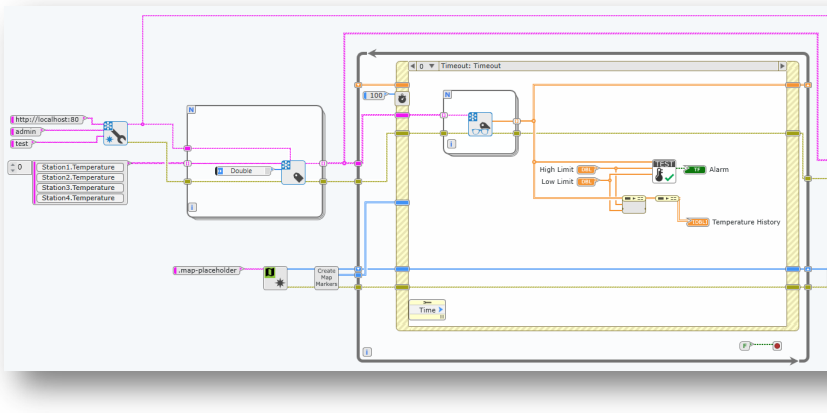


6. Place **Close Tag** from the *Data Communications»SystemLink»Tags* palette and **Close Configuration** from the *Data Communications»SystemLink»Configuration* palette. Connect the wires as shown in the image.



Note that technically in our case, these Close functions will never be called because our While Loop never stops. We include these instructions simply to show the entire API flow. You can choose to wire a Stop button to your While Loop, but you will need to show on the Panel that the WebVI is stopped using a method of your choosing (such as a Boolean button), or else your end user will not know that your WebVI is not functioning.

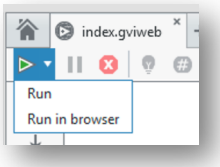
7. Your diagram should now look like the completed diagram below.



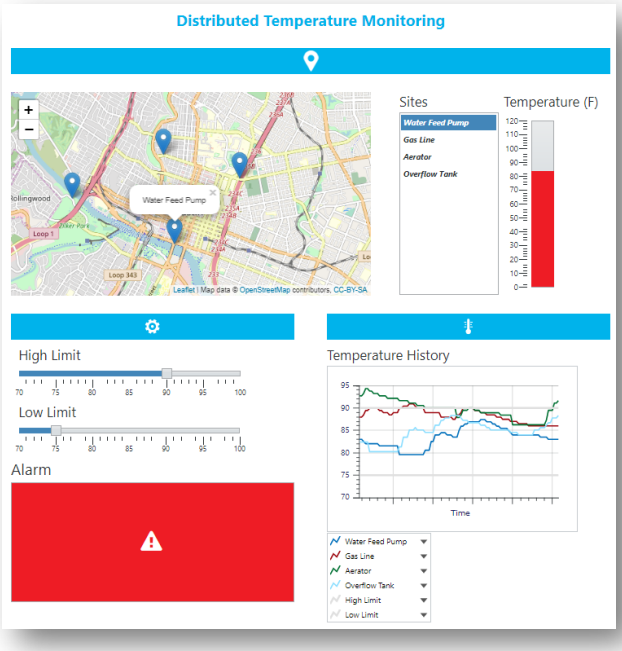
Part G: Run the Temperature Monitoring Web App

Run and Explore the WebVI

5. To run the WebVI, you can choose to either **Run** the WebVI inside LabVIEW NXG, or you can choose to **Run in browser** which will build your WebVI and automatically open the local build in your default browser.



6. Click different locations in the Sites list on the front panel to display different tag data from the server. The site is highlighted on the map.



7. View how the temperature changes on the Temperature History graph.
   Note that when the temperature exceeds the limits, the Alarm indicator turns red.

For the sake of time, this is where we will stop today. Know that from here, you can build your web application and host it locally on the NI Web Server, configure the NI Web Server to allow remote access, and therefore effectively host your web application locally instead of in the cloud like we walked through earlier in the guide.

Thank you for participating! Please explore the LabVIEW NXG Web Module more through these resources:

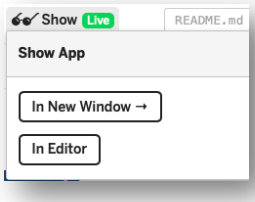
- Webvi.io – this web page is itself an example of a WebVI, and it provides easy access to other WebVI examples
- ni.com/labview/webmodule
- In-product Examples and Lessons
- The white paper *Security in NI Web Technology* which is available online at http://www.ni.com/en-us/innovations/white-papers/18/security-in-ni-web-technology.html

Appendix A – Demo Data Server

For the first exercise, we used the website **glitch.com** to host a simulated temperature data server. National Instruments is not affiliated with glitch.com and does not endorse its use. However, if you want to create your own data server (e.g. to overcome the connection limit on the one created by National Instruments), complete the following steps:

1. In a web browser, navigate to https://glitch.com/@ni
2. The niweek-2019-webvi project will be shown under Featured Project. Click the **Remix your own** button to create a copy of the project.

3. After the project is created and opened, click the *Show* button and choose **In New Window** from the drop-down.



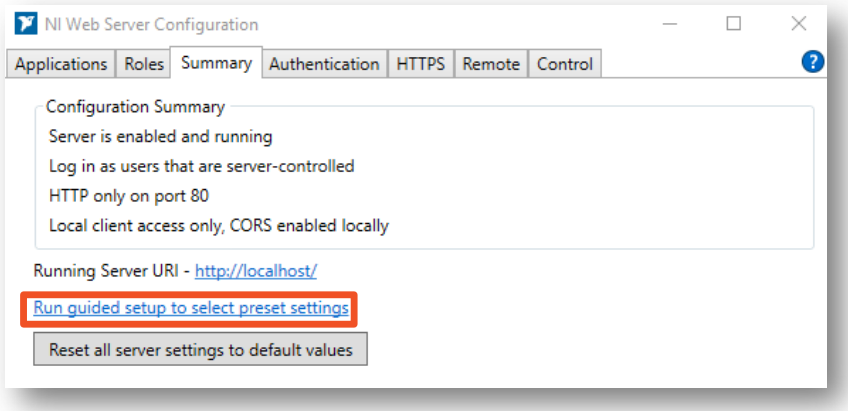
4. Copy the URL of the page that opens and use it instead of **https://niweek-2019-webvi.glitch.me** in the first exercise.

Appendix B – NI Web Server Configuration

Follow these steps to start the NI web server running on your computer and configure it to allow communication with LabVIEW NXG.

1. Open **NI Web Server Configuration**.
    a. On Windows 10: Type "NI Web Server Configuration" into the search box in the task bar and then select it in the search results.
    b. On Windows 7: Navigate to **Start»All Programs»National Instruments»NI Web Server Configuration** via the Windows Start Menu.
2. If the **NI Web Server Guided Setup** window opens, proceed to step 4.
3. If the NI Web Server was previously configured on this machine, you will need to start the guided setup manually from within the NI Web Server Configuration utility. Select the **Summary** tab and

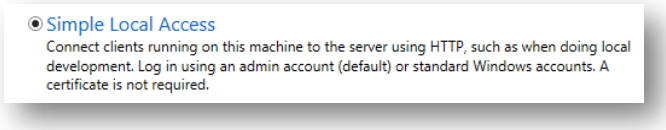
click **Run guided setup to select preset settings**.



Configure the NI Web Server With the Guided Setup Utility

4. Select the **Simple Local Access** option button and click **Next**.

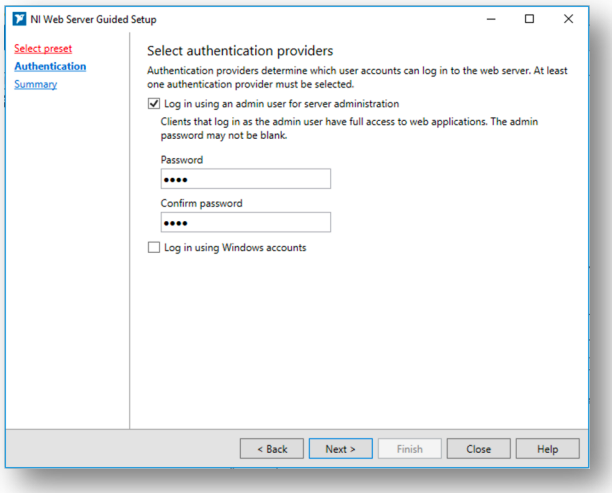
   Simple Local Access allows you to access the web server only from the machine hosting the server. This is useful when developing your web application.



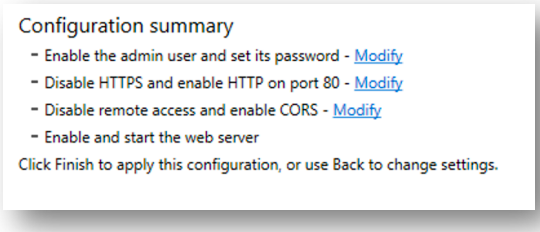
5. **Select Authentication Providers**.

   Place a checkmark in the **Log in using an admin user for server administration** checkbox. This allows you to specify an administrator to configure the server. Create a password for the admin account.

Type *"test"* into the **Password** and **Confirm Password** boxes. For actual applications, NI recommends using a more secure password. Click **Next**.



6. Review the Configuration Summary and verify the summary matches the information below.



Apply and Restart the Web Server

7. Click **Finish** to apply and restart the web server.
   This opens the **NI Web Server Configuration** window.
8. Close the **NI Web Server Configuration** window.

To learn more about the NI Web Server configuration options, please refer to the white paper *Security in NI Web Technology* which is available online at
http://www.ni.com/en-us/innovations/white-papers/18/security-in-ni-web-technology.html