

Siemens
Industry
Online
Support

APPLICATION EXAMPLE

SIMATIC S7-1200 G2 Web Server Guide

SIMATIC STEP 7 (TIA PORTAL V20) / SIMATIC S7-1200 G2

SIEMENS

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. **The application examples are not subject to standard tests and quality inspections of a chargeable product and may contain functional and performance defects or other faults and security vulnerabilities. You are responsible for the proper and safe operation of the products in accordance with all applicable regulations, including checking and customizing the application example for your system, and ensuring that only trained personnel use it in a way that prevents property damage or injury to persons. You are solely responsible for any productive use.**

Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. Any further use of the application examples is explicitly not permitted and further rights are not granted. You are not allowed to use application examples in any other way, including, without limitation, for any direct or indirect training or enhancements of AI models.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.

By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice and to terminate your use of the application examples at any time. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.

The Siemens terms of use (https://www.siemens.com/global/en/general/terms-of-use.html) shall also apply.

Cybersecurity information

Siemens provides products and solutions with industrial cybersecurity functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial cybersecurity concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial cybersecurity measures that may be implemented, please visit www.siemens.com/cybersecurity-industry.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Cybersecurity RSS Feed under https://www.siemens.com/cert.

Table of contents

1. Preface 4

2. Overview of the System 5

2.1. User-defined Web Applications with the S7 Web API 6

3. Application Example 8

3.1. Hardware and Software Components 8

3.2. Installation of the Application Example 10

3.2.1. Configuring the Hardware 11

3.2.1.1. Creating the program blocks and tags in the S7 user program. 13

3.2.2. Download of the Web Application Example onto the S7-CPU 14

4. Web Application 19

4.1. Web Application Structure 19

4.1.1. File Overview 19

4.1.2. Interface Components 19

4.1.3. Communication Flow 20

4.1.4. Customization and Scalability 22

4.1.5. Web API Demo Page 22

5. Appendix 25

5.1. Service and support 25

5.2. Links and literature 26

5.3. Change documentation 27

 © Siemens 2026

1. Preface

Aim of the application example

This guide is designed to help beginners to create a basic web page for the SIMATIC S7-1200 G2 PLC using the S7 Web API and standard HTML editors. It provides a step-by-step introduction for building and deploying custom web applications on the PLC.

Main contents of the application example

This application example highlights the following key points:

- Fundamentals for user-defined web applications
- Creating user-defined web applications with Web API
- Configuring the web server for using user-defined web applications
- Download of a web application Project into the PLC

NOTE

This guide is intended for users without prior experience in web development. The instructions are designed to be clear and accessible, enabling users to create and deploy a basic web application using simple HTML and JavaScript, without the need for advanced technical knowledge.

The application example and the web server should not and cannot replace an HMI system.

2. Overview of the System

Web Server in the SIMATIC controller

Modern automation technology is evolving to integrate internet technologies alongside Ethernet-based communication, which enables direct access to systems through the intranet using standards like HTTPS. This integration allows for flexible evaluation, diagnostics, and control of controllers over large distances, utilizing mobile devices such as tablets and smartphones.

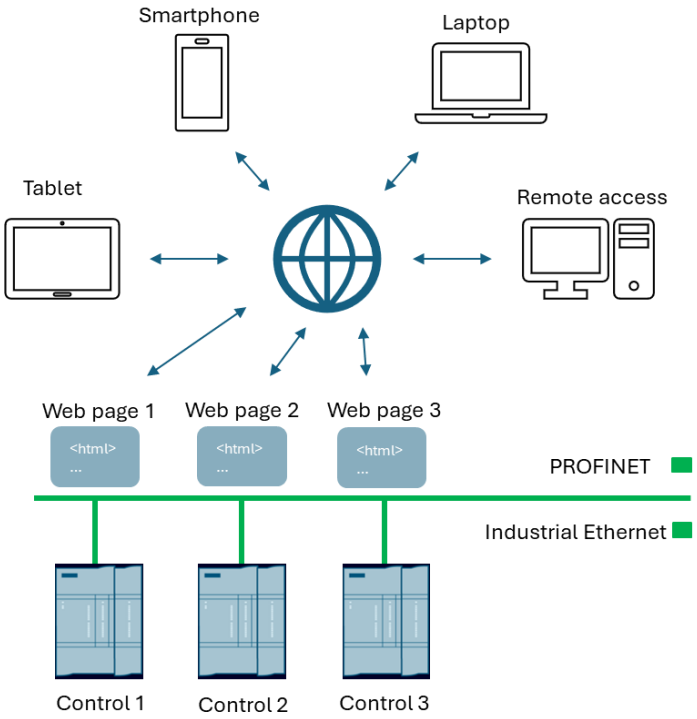


Figure 2-1: Overview of the Automation System and Web Server Architecture

During the test and commissioning phase, engineers require flexible CPU access to visualize individual data during operations for diagnostic purposes. The advantage of accessing control data through various web browsers is that it can be displayed and controlled to a limited degree on any computer or web-enabled device without the need for additional software.

The S7-1200 G2 CPU web server offers the option to create custom web applications tailored to their specific use cases and requirements. This manual is dedicated to guiding you through the creation of user-defined web applications using the Web API.

Advantages of the Web Server

- **Integrated web pages.** User-defined web pages can be individually created and customized.
- **Location-independent.** Web pages can be accessed worldwide via a standard internet browser. Access is possible via mobile communication devices, e.g., tablet PC, smartphone.

Application example. Universal use of the application example for all control systems.

2.1. User-defined Web Applications with the S7 Web API

Currently, the Web API is available for developing custom web applications to connect with the PLC. It can be developed and tested independently from the PLC code.

This document specifically focuses on the development of custom web pages using the Web API and web applications. To create your user-defined web page with Web API, you can use tools such as Visual Studio Code or Notepad++. For designing your web page, you can use all possibilities provided to you by HTML and JavaScript.

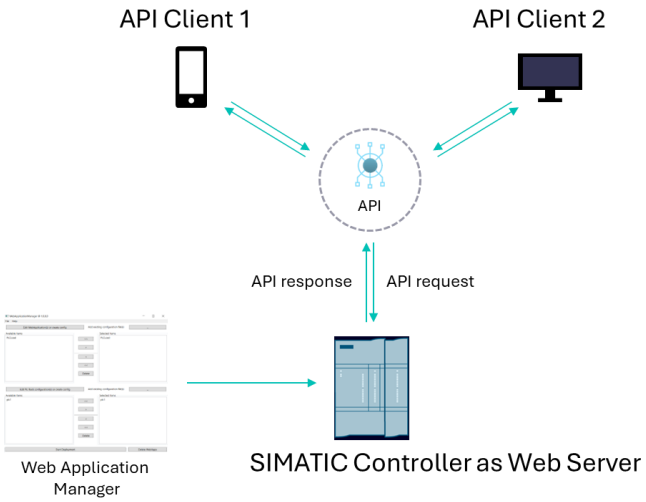


Figure 2-2: Architecture of the web application: Communication between engineering station, web browser, and S7-1200 G2 via PROFINET

Thanks to the Web API and the Web Application Manager, the creation of a web application and its transfer is completely independent from the development of the PLC program in the TIA Portal. Once the location of the process data in the CPU (Data Block) is defined, the web developer can begin the development process.

The following steps describe the procedure for creating user-defined web pages:

PLC Program Development

- Use SIMATIC STEP 7 (TIA Portal) to develop the PLC program, ensuring it includes program code, hardware configuration, User Management and Web Server activation.
- Download all program blocks onto the CPU to ensure proper implementation.

Web Page Development

Create the HTML file for the user-defined web page and link it to the PLC using the Web API.

Web Project Download onto the PLC

Use the Web Application Manager to download the web project to the CPU.

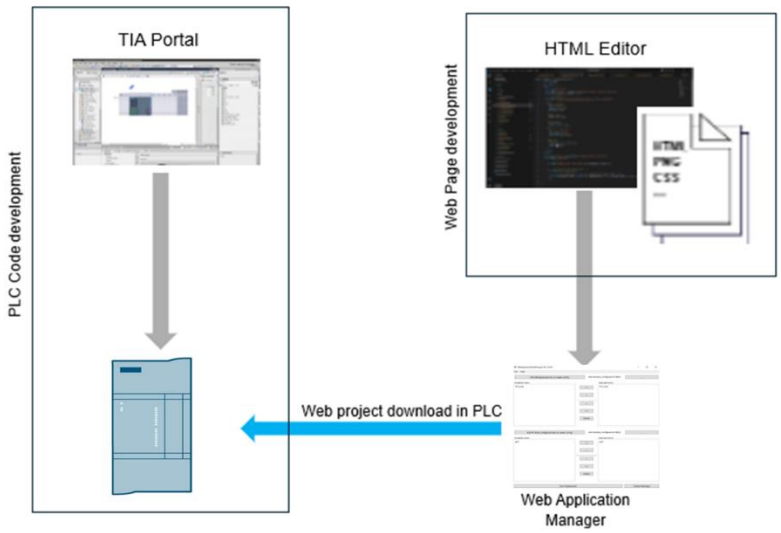


Figure 2-3: Procedure of creating user-defined web pages

3. Application Example

This application example demonstrates the development of a custom web project using plain HTML and JavaScript, supported by the jQuery library. It showcases how to interact with the S7-1200 CPU by leveraging the Web API methods to access tags and parameters. The implementation focuses on integrating the Web API with a lightweight front-end, enabling dynamic communication between the web interface and the PLC.

3.1. Hardware and Software Components

This application example was implemented with a S7-1200 G2 CPU.

A computer serves for the creation of the PLC project and the web content, as well as for loading them to the CPU and displaying the web pages in a web browser.

The application example was created with the following components:

Hardware Components

Component	Item number	Note
SIMATIC S7-1200 G2: compact CPU 1212C DC/DC/DC	6ES7 212-1AG50-0XB0	FW V1.0

Table 3-1

Software Components

Component	Item number	Note
SIMATIC STEP 7 Professional V20	6ES7822-1AE24-0YA5	-
Software for creating HTML files and scripts, e.g., Visual Studio Code	-	-
Web Application manager	simatic-s7-webserver-api/samples/WebAppManager at main · siemens/simatic-s7-webserver-api · GitHub	Tool used to download the Web application onto the S7-CPU
Web browser, e.g., Google Chrome, Microsoft Edge	-	-

Table 3-2

The following web browsers were explicitly tested for communication with the CPUs:

- Google Chrome
- Microsoft Edge

Content of the application example

You will find the following in-depth content in the application example:

- Configuration of the web server for a CPU S7-1200 G2
- Creation of a user-defined web page for the CPU with the following functions:
  - Displaying CPU tags.
  - Setting of CPU tags.
  - Login form to access to CPU.
- Particularities in the TIA Portal user program creation:
  - Providing tags for the web page.

Code of the Application Example

Components	Comments
Project for TIA Portal V20	A project in TIA Portal specifically designed for TIA Portal version 20.
Custom web page	The zip file contains the corresponding scripts and html files needed in the web application. The web application files are in an independent folder called “BasicWeb”.
Web API demo page	An interactive page for exploring and testing Web API methods.

Table 3-3

3.2. Installation of the Application Example

Hardware installation

The computer with the web browser must be connected to the CPU via Industrial Ethernet. There are multiple ways to do it, but the simplest solution is to directly connect it with one of the PN interfaces of the CPU.

NOTE	Please observe the installation and connection guidelines from the corresponding manuals.

Installing the software

1. Install SIMATIC STEP 7 (TIA Portal).
2. Install a web page creation tool on the computer that you want to create the web page. It is recommended to have an IDE (integrated development environment) to facilitate the use of third-party software packages and other benefits like the developer tools, integrated terminal, compiler, etc. For example, Visual Studio Code.
3. Install a web browser such as Google Chrome or Microsoft Edge on the computer with which you want to access the web page on the CPU.
4. Install the Web Application Manager on the CPU to download the web application to the PLC.

Procedure for Creating a Web Page

The configuration and settings in STEP 7 and the writing of the HTML file are closely related. The following procedure is recommendable for that:

1. Configuring the Hardware and the Web Server settings.
2. Creating the program blocks and tags in the TIA Portal user program.
3. Compile and download the user program.

Thanks to the Web API and the Web Application Manager, the creation of the web application and its transfer is completely independent from the development of the user program. Once it is defined where the data is going to be located (definition of the data blocks), the developer can start the process:

4. Creating the HTML files and scripts.
5. Download the web application with the Web Application Manager

3.2.1. Configuring the Hardware

1. Start the TIA Portal and select "Project > New..." to create a new project with the name "WebServer_S71200_G2", for example.
2. Add a S7-1200 G2 station via "Add new device > Controller > SIMATIC S7-1200 G2". The Device View of the CPU opens.
3. Assign the IP address 192.168.0.1 of the CPU to PROFINET Interface X1. Via this IP address, you will later use the web browser to access the web page loaded on the CPU.

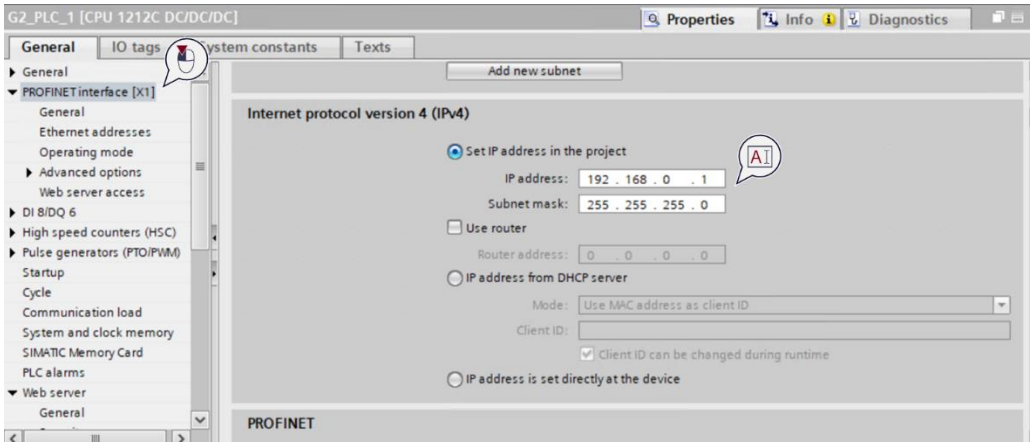


Figure 3-1: PLC Configuration

4. Click the "Web Server Access" option in the properties of the PROFINET Interface. Enable web server via IP address of this interface.

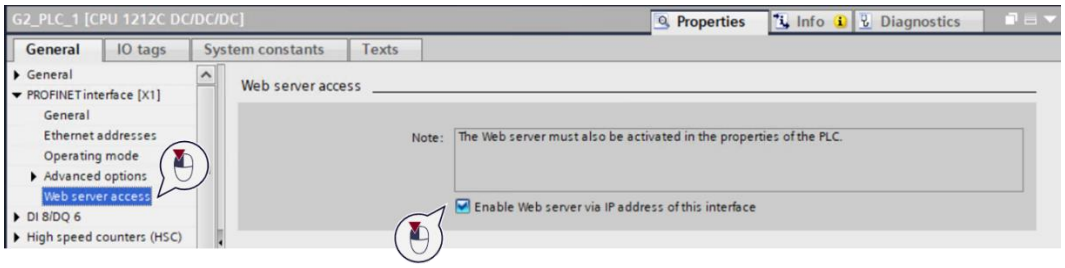


Figure 3-2: PLC Configuration, web server access

5. Click on "Web server" under the General configuration of the PLC and select activate web server on this module.

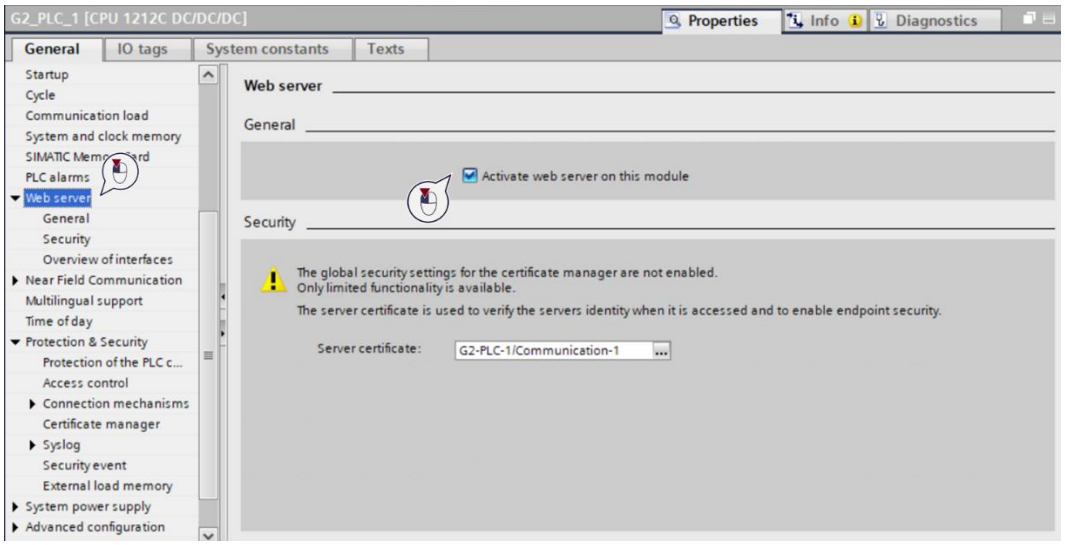


Figure 3-3: PLC configuration, web server configuration

6. On the project tree configuration select: Security Settings > Settings > Project Protection. Click on "Protect this project". This will enable some security features like for example the certificate manager where you can download the CA certificate and device certificate. Once a project is protected it can't be unprotected anymore.

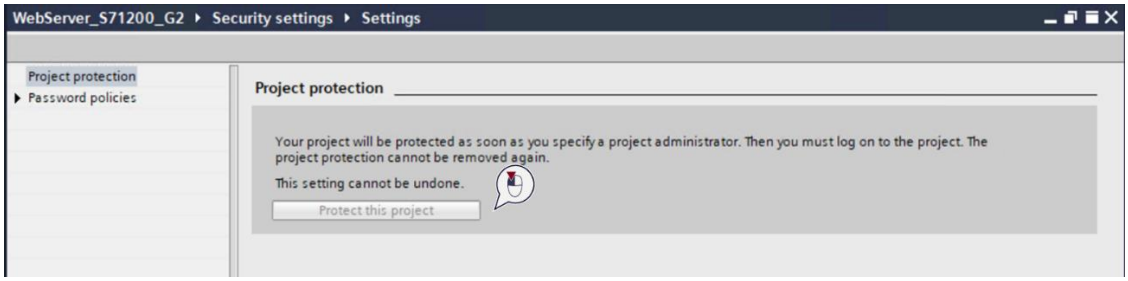


Figure 3-4: Project configuration, protection

7. Enter the following login data and confirm with "OK":
   a. Username: UserAdministrator
   b. Password: Siemens123!
8. Go to Users and roles and there you will find the previously created user. Now create a new user with the following data:
   a. Username: User
   b. Password: Siemens123!



Figure 3-5: PLC Configuration, users

9. Go to Roles and create a new "WebServer" role. The new role has the function rights for each PLC of the project: S7-1200 G2

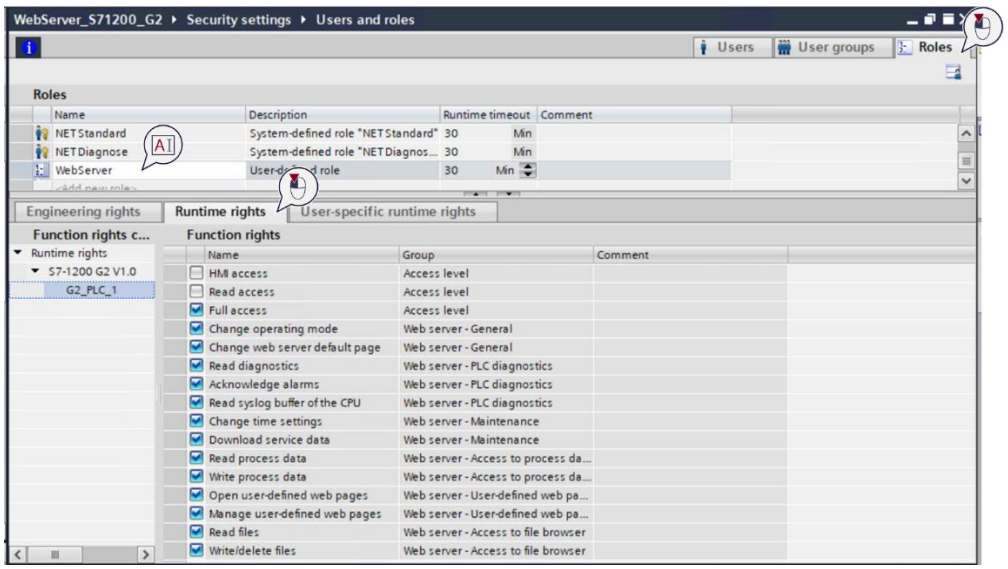


Figure 3-6: PLC Configuration, WebServer role

10. Go to Users and assigned the role “WebServer” to User

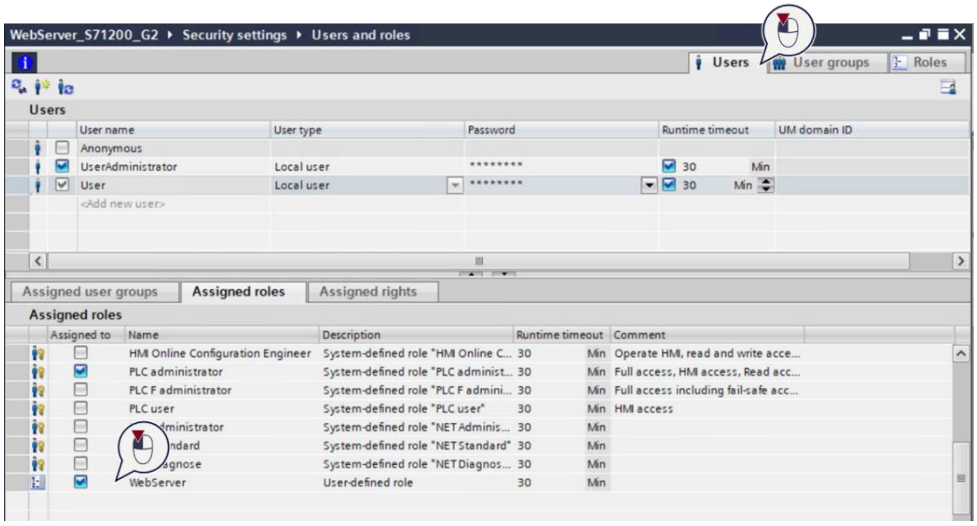


Figure 3-7: PLC Configuration, Assigned role

3.2.1.1. Creating the program blocks and tags in the S7 user program.

As part of this application example, a dedicated Data Block (DB) is created within the TIA Portal project to define and store two variables. These variables are intended to be accessed and modified via the user-defined web interface using the Web API.

It is important to note that this implementation serves solely as a demonstration of the interaction between the web application and the PLC. The creation of such a DB is not mandatory and may vary depending on the specific requirements of the user’s automation project.

1. Create the Global DB “DataFromWeb”

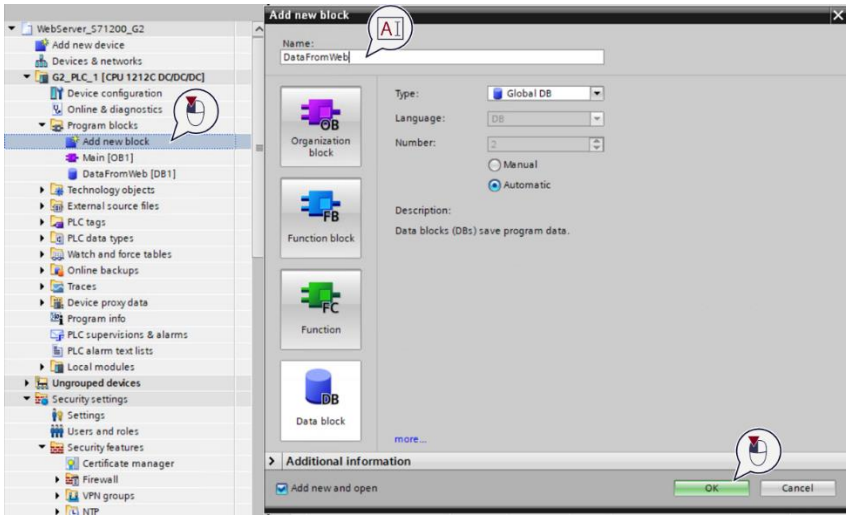


Figure 3-8: Program blocks, Add new Block

2. The relevant variables must be defined and configured within the Data Block. This configuration enables external access to the tags, allowing read and write operations from the web interface.

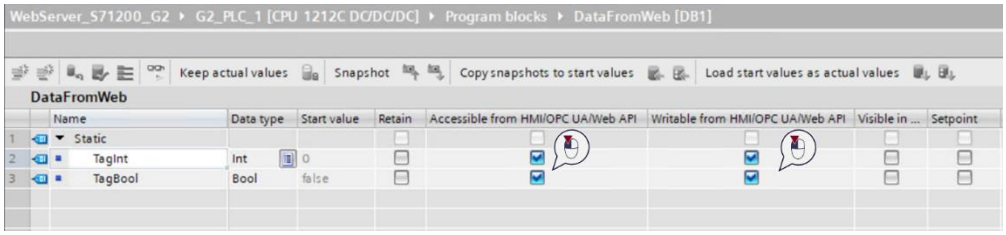


Figure 3-9: Program blocks, Tags definition

3.2.2. Download of the Web Application Example onto the S7-CPU

To download the application example into the PLC you need to start the Web Application Manager. This application is an open-source application that allows a user to simply perform the download of a complete web application to the PLC, making the process completely independent from TIA Portal.

Reference to repository: simatic-s7-webserver-api/samples/WebAppManager at main · siemens/simatic-s7-webserver-api · GitHub

Here you can download an executable that contains all the necessary files to run the application

How to download an application to the PLC

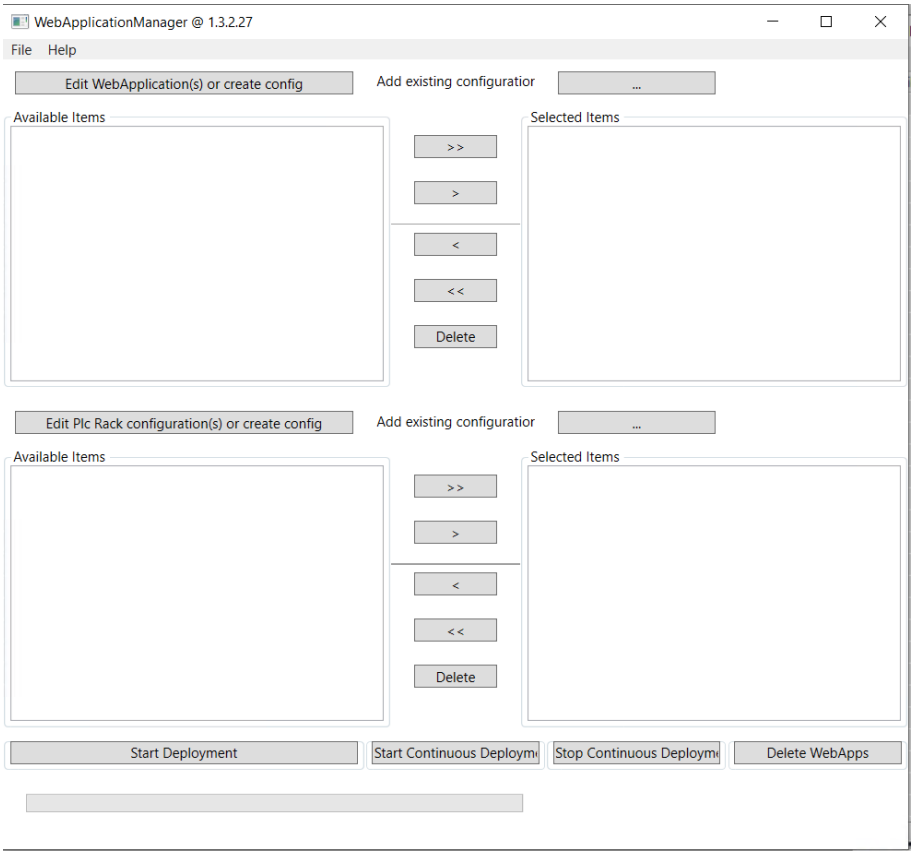


Figure 3-10: Web Application Manager interface

1. Edit Web Application(s) or create config: Create a configuration for the web application to be downloaded.

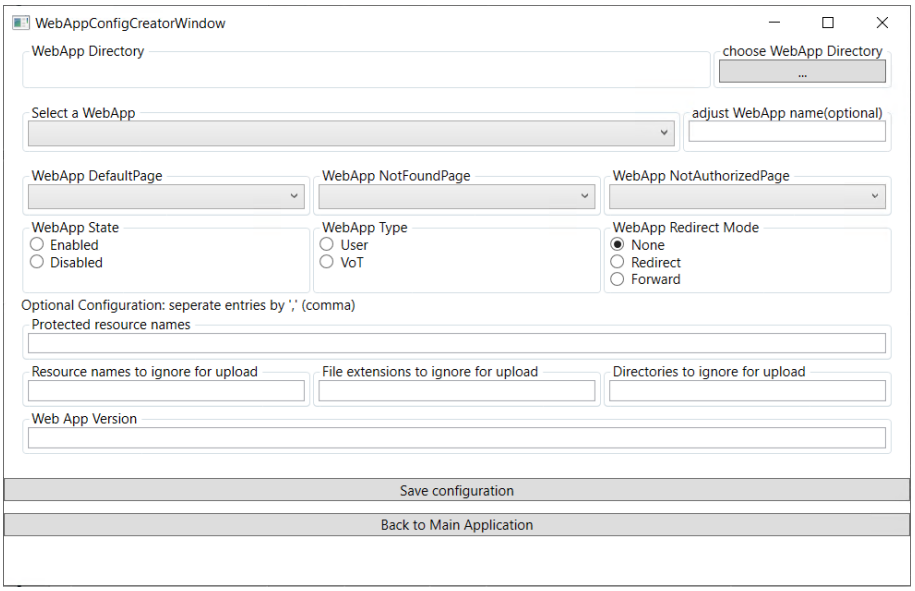


Figure 3-11: Web Application Manager configurator creator window.

2. The minimum configuration that needs to be done here is:
   a. Choose a WebApp Directory
   b. Select a WebApp: Select the folder where all the content of the web app is stored. It automatically will maintain the subfolder structure.
   c. Select a DefaultPage.
   d. WebApp State: Enabled (if you want users to access it).
   e. WebApp Type: User.
   f. Save the configuration and go back to the main application

   For this application example:

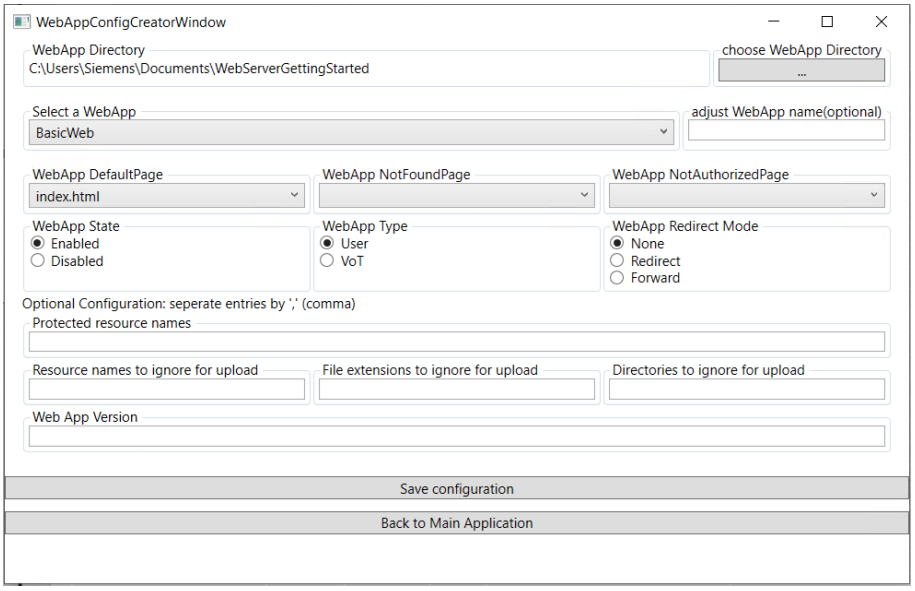


Figure 3-12: Web Application Manager configuration for this application

3. Edit Plc Rack configuration(s) or create config: Create a configuration for the PLC or rack of PLCs where you want to download the application. Proceed as follows:
   a. First add a name to the Plc rack and then click on Add button.

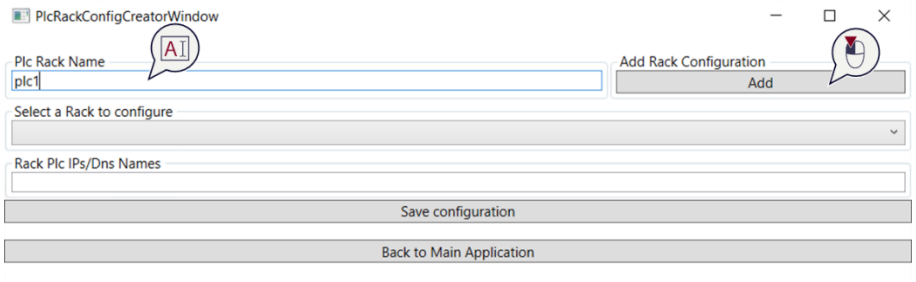


Figure 3-13: Web Application Manager PLC rack configuration creator window

b. The next step will be to select the rack from the list of racks and give it the list of IPs that you want to address separated by commas. For example:

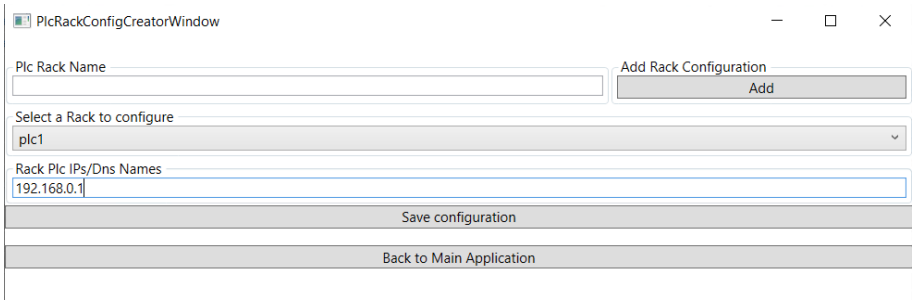


Figure 3-14: Web Application Manager PLC rack configuration for this application

c. Save the configuration and go back to the main application.

4. Now is time to perform the download of the application(s) to the PLC(s):

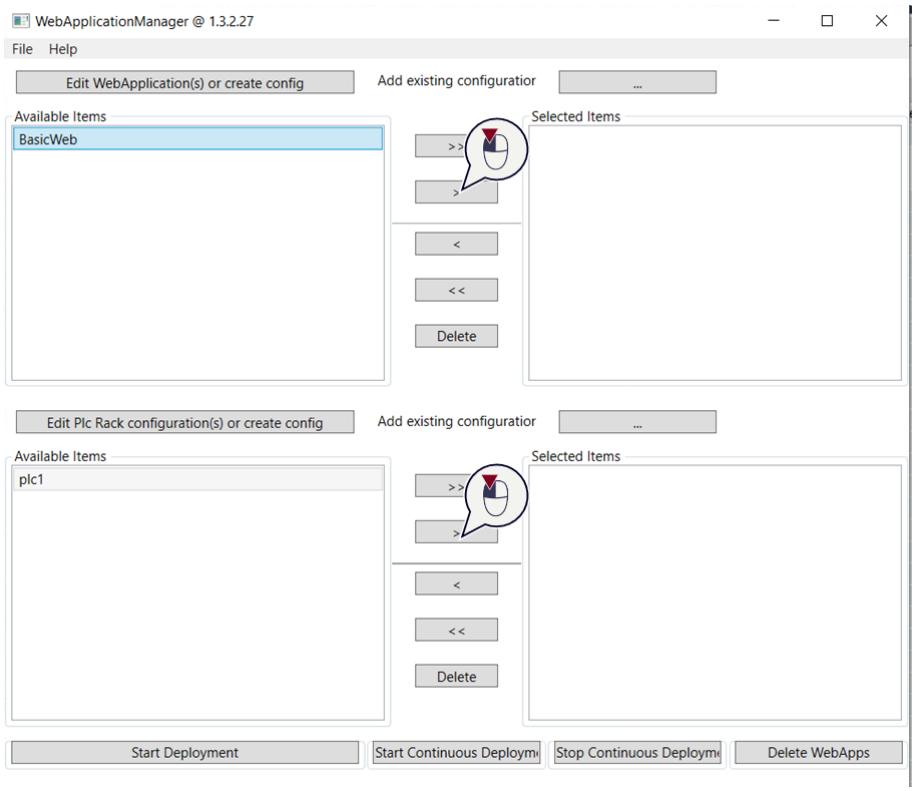


Figure 3-15: Web Application Manager Download Process 1

What is going to be transferred and the place that will be directed needs to be on the right side of the main application. To do that, select the objects on the available items list, click on the ">" symbol. The result will be:

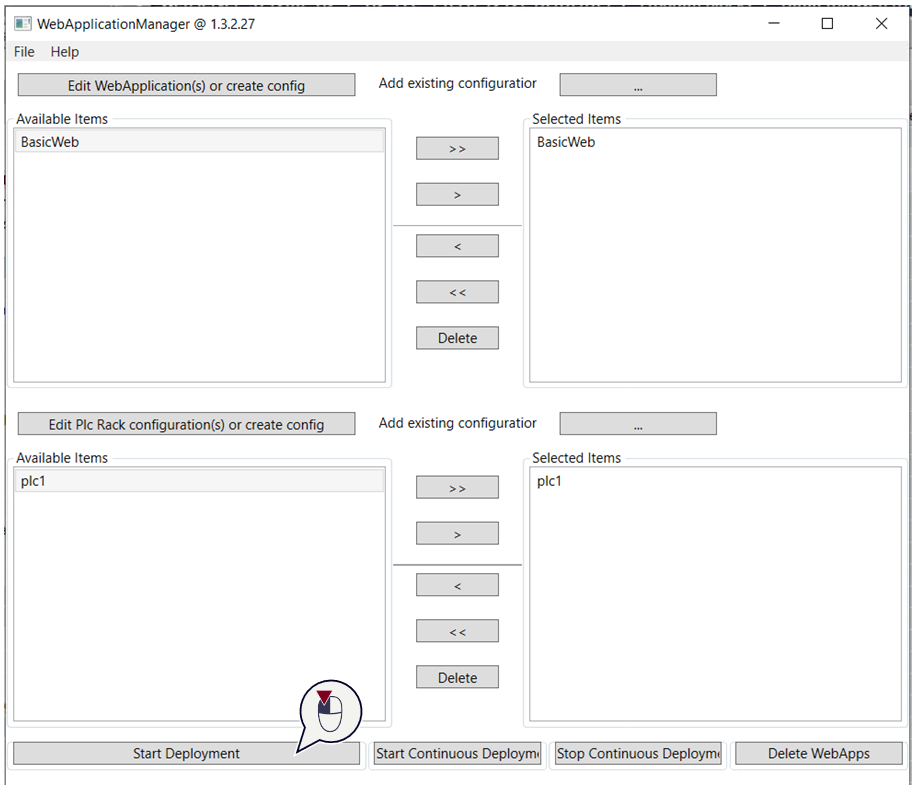


Figure 3-16: Web Application Manager Download Process 2

Now you can click on Start Deployment. You will be redirected to a form that will request a user and password with enough rights (manage user defined web pages is mandatory):

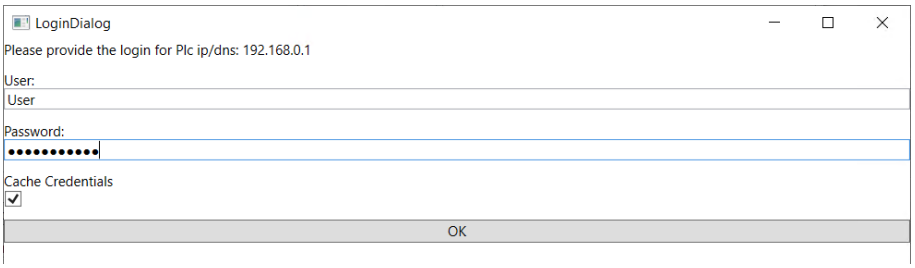


Figure 3-17 : Web Application Manager Login dialog

Clicking on OK will start the download process automatically.

NOTE Note that if an application has been previously downloaded to a PLC it is possible also to delete it from here following the same procedure for downloading but clicking on the "Delete WebApps" button.

NOTE It is possible to select multiple PLC racks and multiple web applications to perform a batch download to different rack configurations with a single click.

NOTE The maximum number of user-defined web applications that a PLC can host at the same time is four.

Users and Passwords

Application	User	Password
TIA Portal – Open Project	UserAdministrator	Siemens123!
TIA Portal – Download Project	User	Siemens123!
User Defined Web Page	User	Siemens123!

Table 3-4

How to access the Web page

Open the user-defined page as follows:

1. Start a web browser, such as Google Chrome.

   For the address enter, the IP address of the CPU, plus a "~", plus the name of the application, in this case: "BasicWeb ", e.g. https://192.168.0.1/~BasicWeb

2. The "Login" web page opens.

   Now the user is requested to perform a Web API session by logging in with the username and password. In this case the user used is the same as the previous one, with the following login data:

   - Username: User
   - Password: Siemens123!

3. Click the Login button. If the session starts successfully, the Token will be visible in the corresponding field.

4. Web Application

4.1. Web Application Structure

The web application is designed to provide a lightweight and functional interface for interacting with the S7-1200 CPU via the Web API. It is implemented using plain HTML and JavaScript, with jQuery as the supporting library for asynchronous communication and DOM manipulation. The structure is modular and consists of the following key components:

4.1.1. File Overview

index.html

HTML (HyperText Markup Language) is the standard language used to create and structure content on web pages. It defines elements such as text, buttons, input fields, and layout containers.

This file is the main HTML document that defines the structure and layout of the user interface. It includes:

- A login form for user authentication.
- A control panel for reading and writing PLC variables.
- References to external scripts and style sheets.
- Responsive design settings via meta tags.

jquery-3.7.1.min.js

jQuery is a lightweight JavaScript library that simplifies tasks such as sending HTTPS requests and manipulating HTML elements. It is used here to communicate with the S7 Web API using GET and POST methods.

script_test.js

JavaScript is a scripting language that enables dynamic behavior on web pages. In this application, it is used to send and receive data from the PLC, update the interface, and handle user interactions.

The file contains the core JavaScript logic for the application, including:

- User authentication via the Web API.
- Reading and writing tag values.
- Handling UI events and updating the interface dynamically.
- Error handling and response validation.

WebAppConfig.json

A configuration file that stores application parameters used by the Web Application Manager.

4.1.2. Interface Components

The application interface is divided into functional sections:

Login Panel

Allows users to enter credentials and obtain a session token. The token is required for authenticated access to the Web API. The login form includes:

- Username and password fields.
- A token display area.
- Login and logout buttons.

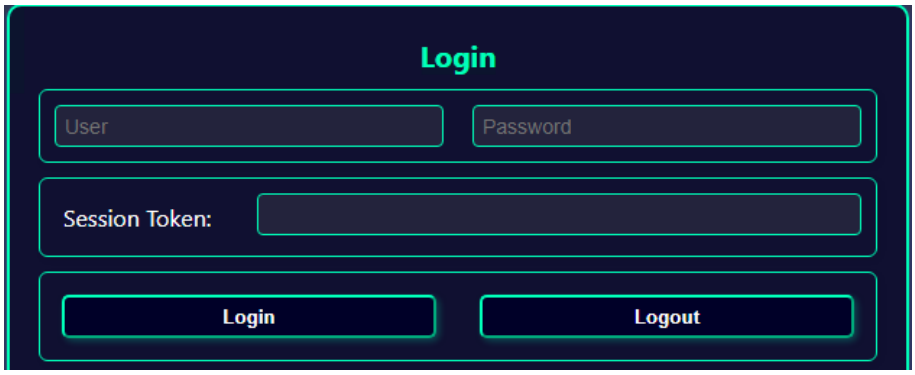


Abbildung 4-1: Web application login

Control Panel

Provides controls for interacting with PLC variables. It includes:

- Boolean control section with ON/OFF buttons. Connected to the "TagBool" variable inside the "DataFromWeb" block
- Integer control section with sliders and numeric input fields. Connected to the "TagInt" variable inside the "DataFromWeb" block
- Read and write buttons for each variable group.
- Visual feedback for current values and operation status.

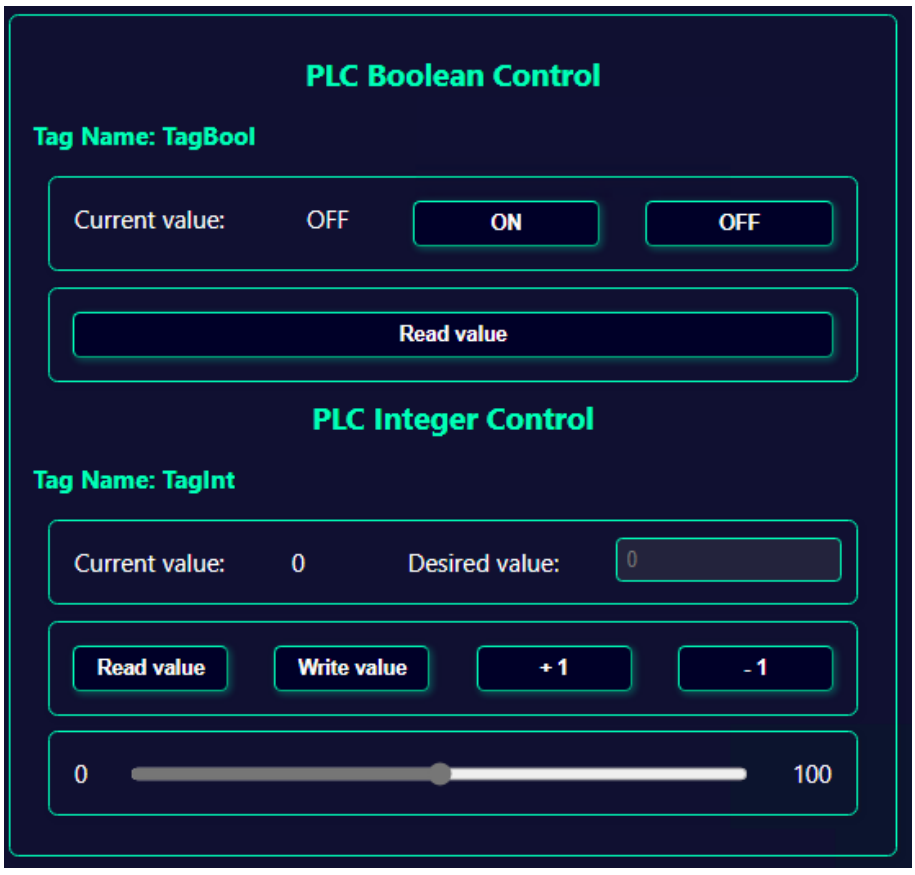


Abbildung 4-2: Web application, Control Panel

4.1.3. Communication Flow

Communication Model: Request–Response via Web API

The web application communicates with the S7-1200 CPU using the HTTPS-based request–response model. As illustrated in the figure below, the client (web browser) sends a **request** to the server (PLC), and the server processes the request and returns a **response**.

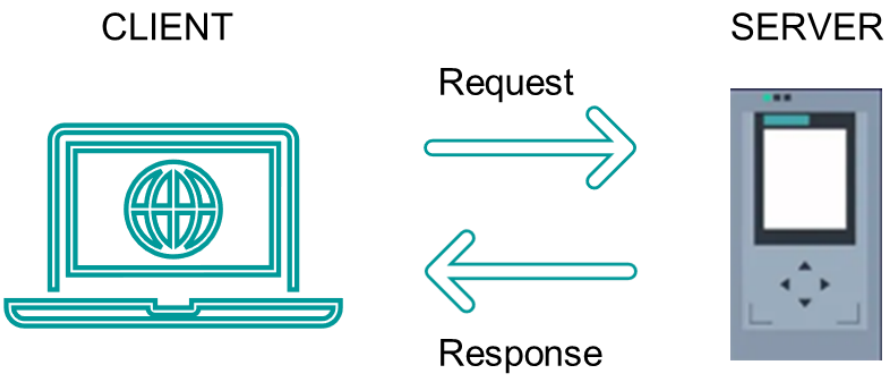


Figure 4-3: Request–Response communication model between the web client and the S7-1200 Web

This interaction is fundamental to the operation of the Web API.

Each request includes necessary parameters such as the tag name, data type, and session token (if authentication is enabled). The server responds with a JSON object containing the result of the operation, which is then processed and displayed by the web interface.

This model ensures a clear and structured communication flow, enabling real-time monitoring and control of PLC variables from a standard web browser.

The following JavaScript function demonstrates the login process using the Web API. It is divided into two main stages: Request Construction and Transmission, and Response Handling.

1. **Request: Building and Sending the Login Request**

```
let request = JSON.stringify({
    "jsonrpc": "2.0",
    "method": "Api.Login",
    "params": {
        "user": user,
        "password": password
    },
    "id": requestCounter++
});
$.post({
    url: "https://192.168.0.1/api/jsonrpc",
    data: request,
    ...
});
```

- **User Input**: The function retrieves the username and password entered by the user in the login form.
- **Request Format**: The request is structured using the JSON-RPC 2.0 protocol, specifying the method Api.Login and passing the credentials as parameters.
- **Request ID**: A unique identifier (id) is included to match the request with its corresponding response.
- **Transmission**: The request is sent via an HTTP POST to the Web API endpoint of the S7-1200 CPU.

2. **Response: Handling the Server Reply**

```
success: function(data) {
let parsed = JSON.parse(data);
sessionToken = parsed.result.token;
document.getElementById("token").textContent = "Token: "+ sessionToken;
document.getElementById("controlPanel").style.display = "block";
```

```
},
error: function(xhr, status, error) {
alert("Login fail");
console.error("Error:", status, error);
},
```

- **Success Callback**: If the login is successful, the server returns a JSON-formatted response containing a session token.
    - The token is extracted and stored in the sessionToken variable.
    - It is displayed in the interface and used to authorize subsequent API requests.
    - The control panel becomes visible, enabling user interaction with PLC variables.
- **Error Handling**: If the login fails, an alert is shown, and the error is logged to the console for debugging.

This mechanism ensures secure and structured communication between the web application and the PLC, enabling authenticated access to protected resources.

4.1.4. Customization and Scalability

The application is designed to be easily customizable:

- Additional tags can be added by modifying the configuration file and extending the HTML interface.
- Styling can be enhanced using CSS.
- The logic can be extended to support more data types or advanced control features.

This modular and transparent structure makes the application suitable for educational purposes, prototyping, and integration into larger automation systems.

4.1.5. Web API Demo Page

To facilitate the exploration and testing of the available Web API methods, a dedicated Web API Demo Page is provided. This local HTML page serves as an interactive tool, allowing users to understand the structure of each method, input parameters, and observe the resulting JSON requests and responses without needing to write code from scratch.

The demo page is intuitively structured to guide through the API's functionalities:

Settings Section (Top of the Page):

This section enables the configuration of essential connection parameters for interacting with the PLC's Web API.

- **IP address**: The IP address of the PLC is entered here. The "Go to" link provides a quick way to navigate to the PLC's integrated web pages.
- **API Session Token**: Session token required for authentication.
- **Keep user session alive**: A checkbox to manage session persistence.
- **User session alive time (seconds):** Defines how long the user session should remain active.
- **Omit X-Auth-Token from API requests:** An option to control the inclusion of the X-Auth-Token.
- **R/H PLCs: Redundancy ID:** For redundant PLC setups, this field allows specifying the redundancy ID.

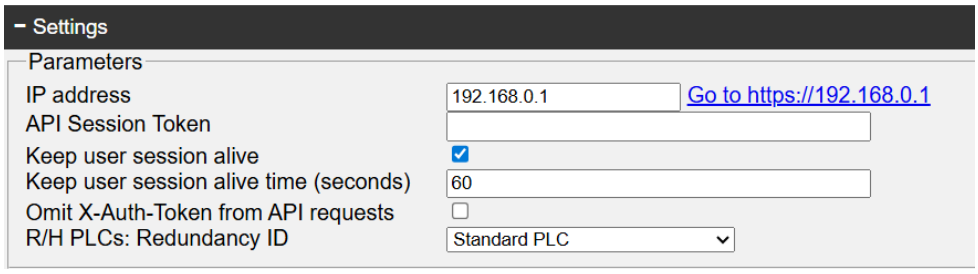


Figure 4-4: Settings Section

API Methods Section (Main Body):

This is the core of the demo page, listing all available Web API methods, categorized for easy navigation.

Methods are grouped by functionality (e.g., Api.*, Alarms.*, DataLogs.*, Plc.*, Project.*, WebApp.*, WebServer.*).

Each category is an expandable section (indicated by a + sign). Clicking on a category reveals the specific methods within that group.

Exploring a Specific Method (Example: Api.Login)

When an API method category is expanded and a specific method is selected (e.g., Api.Login as shown in the example), a detailed interactive panel appears, structured into three main parts:

Request Section:

This area facilitates the input of necessary parameters for the selected API method.

Elements:

- Parameters: Each parameter required by the method (e.g., user, password, mode for Api.Login) is listed with its type and whether it is mandatory or optional.
- Input Fields: Corresponding input fields (text boxes, dropdowns, checkboxes) are provided for parameter value entry.
- Call Method Button: Once all desired parameters are set, clicking this button executes the API call.

JSON Request Section:

This read-only section dynamically displays the JSON payload that will be sent to the PLC's Web API when the Call Method button is pressed.

It is useful for understanding the exact JSON structure and content that corresponds to the provided inputs, which is crucial for debugging and for implementing these calls in applications.

JSON Response Section:

After the Call Method button is clicked, the response received from the PLC's Web API is displayed here in JSON format.

This provides immediate visibility of the API call's result, including any data returned or error messages, aiding in the confirmation of the method's behavior and expected output.

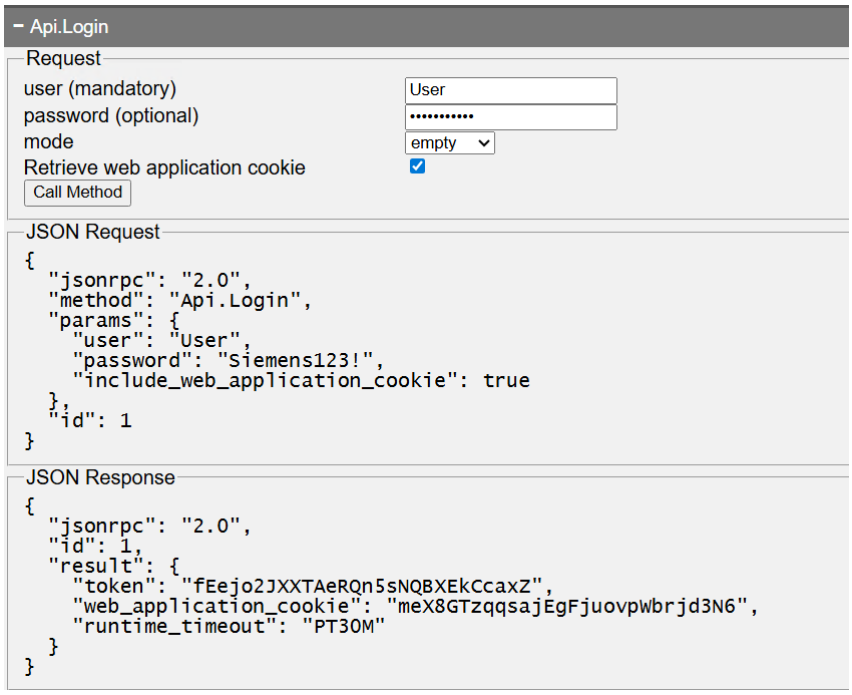


Figure 4-5: Example: Api.Login

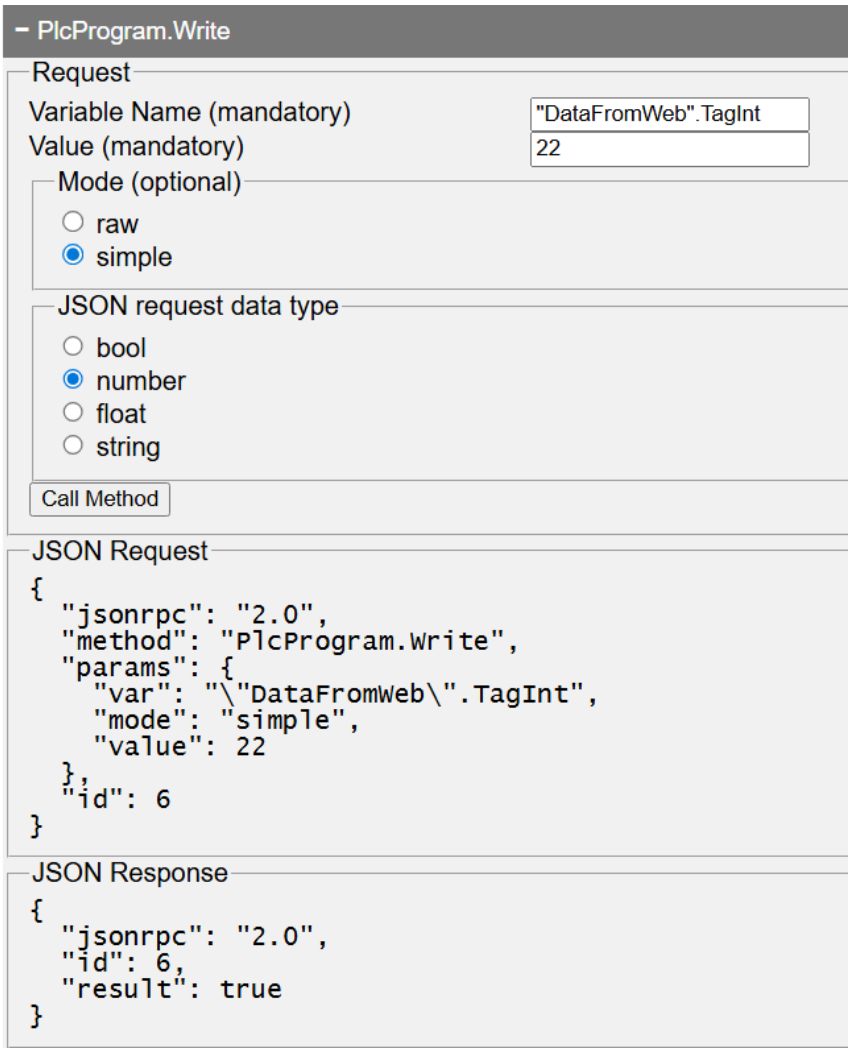


Figure 4-6: Example: PlcProgram.Write

5. Appendix

5.1. Service and support

SiePortal

The integrated platform for product selection, purchasing and support - and connection of Industry Mall and Online support. The SiePortal home page replaces the previous home pages of the Industry Mall and the Online Support Portal (SIOS) and combines them.

- Products & Services
  In Products & Services, you can find all our offerings as previously available in Mall Catalog.
- Support
  In Support, you can find all information helpful for resolving technical issues with our products.
- mySiePortal
  mySiePortal collects all your personal data and processes, from your account to current orders, service requests and more. You can only see the full range of functions here after you have logged in.

You can access SiePortal via this address: sieportal.siemens.com

Industry Online Support

Industry Online Support is the previous address for information on our products, solutions and services.

Product information, manuals, downloads, FAQs and application examples - all information is available with just a few mouse clicks: support.industry.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers – ranging from basic support to individual support contracts. Please send queries to Technical Support via Web form: support.industry.siemens.com/cs/my/src

SITRAIN – Digital Industry Academy

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that's tailored to the customer's specific needs.
For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page: siemens.com/sitrain

Industry Online Support app

You will receive optimum support wherever you are with the "Industry Online Support" app. The app is available for iOS and Android:

5.2. Links and literature

Nr.	Thema
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to this entry page of this application examplehttps://support.industry.siemens.com/cs/ww/en/view/111
\3\	HTML5, JavaScript https://www.w3schools.com/html/ https://www.w3schools.com/js/
\4\	SIMATIC S7-1200 G2 Programmable Controller, System Manual https://support.industry.siemens.com/cs/ww/en/view/109972011
\5\	SIMATIC STEP 7 Basic/Professional V20 and SIMATIC WinCC V20 https://docs.tia.siemens.cloud/
\6\	HTML and CSS, Practical Formulas for Beginners Robert R. Agular Mitp ISBN 978-3-8266-1779-9
\7\	HTML manual Stefan Münz / Wolfgang Nefzger Franzis Verlag (Publishing house) ISBN 3-7723-6654-6
\8\	JavaScript and Ajax, the comprehensive manual Christian Wenz Galileo Press ISBN 978-3-8362-1128-4

Table 5-1

5.3. Change documentation

Version	Date	Modification
V1.0	12/2025	First version

Table 5-2