SIEMENS

SIMATIC

S7-1500, ET 200SP, ET 200pro Web server

Function Manual

Introduction 1

Safety instructions 2

Industrial cybersecurity 3

Setting up and calling a web server 4

System web pages 5

API (Application Programming Interface) 6

11/2024
A5E53797648-AA

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens Aktiengesellschaft. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens Aktiengesellschaft
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E53797648-AA
Ⓟ 10/2024 Subject to change

Table of contents

1 Introduction 7

1.1 Function Manuals documentation guide 9
1.1.1 Information classes Function Manuals 9
1.1.2 Basic tools 11
1.1.3 SIMATIC Technical Documentation 13

2 Safety instructions 16

2.1 General security information 16

3 Industrial cybersecurity 17

3.1 Cybersecurity information 17
3.2 Cybersecurity-relevant information in this manual 18

4 Setting up and calling a web server 19

4.1 Properties of the Web server 19
4.2 Configuring the Web server 22
4.3 System web pages and previous web pages 26
4.4 Certificates 29
4.4.1 Web server certificates 29
4.4.2 Managing certificates via TIA Portal 30
4.4.3 Managing certificates in runtime 36
4.4.4 Installing certificates for the web browser 38
4.5 Language settings 39
4.6 User management 41

5 System web pages 50

5.1 Getting started 50
5.1.1 Authentication 53
5.1.2 Operating state and messages 58
5.2 Overview 58
5.3 Diagnostics 59
5.3.1 Diagnostics buffer 60
5.4 User program 62
5.5 Alarming & Logging 64
5.5.1 Alarms 65
5.5.2 Data logs 66
5.6 Maintenance 67

5.6.1 File browser ... 67

6 API (Application Programming Interface) ... 71

6.1 Supported clients ... 72
6.2 Web API integration ... 79
6.3 Web API sessions ... 82
6.4 Web API basic functions ... 84
6.4.1 Api.Login ... 85
6.4.2 Api.Logout ... 89
6.4.3 Api.GetPermissions ... 89
6.4.4 Api.ChangePassword ... 91
6.4.5 Api.GetPasswordPolicy ... 92
6.4.6 Api.GetAuthenticationMode ... 94
6.4.7 Api.GetSessionInfo ... 95
6.4.8 Api.Browse ... 98
6.4.9 Api.Version ... 99
6.4.10 Api.Ping ... 99
6.4.11 Api.GetCertificateUrl ... 100
6.4.12 Api.GetQuantityStructures ... 100
6.5 Ticket mechanism ... 101
6.5.1 Api.BrowseTickets ... 106
6.5.2 Api.CloseTicket ... 108
6.6 User-configurable HTTP response headers ... 109
6.6.1 Reading and writing user-configurable HTTP response headers ... 109
6.6.2 WebServer.ReadResponseHeaders ... 112
6.6.3 WebServer.ChangeResponseHeaders ... 114
6.7 Setting the Web server default page ... 115
6.7.1 WebServer.SetDefaultPage ... 115
6.7.2 WebServer.ReadDefaultPage ... 117
6.8 Web applications that can be loaded by the user ... 117
6.8.1 Interaction between web applications ... 123
6.8.2 WebApp.Create ... 125
6.8.3 WebApp.Delete ... 126
6.8.4 WebApp.Rename ... 127
6.8.5 WebApp.Browse ... 128
6.8.6 WebApp.SetState ... 131
6.8.7 WebApp.SetDefaultPage ... 132
6.8.8 WebApp.SetNotFoundPage ... 133
6.8.9 WebApp.SetNotAuthorizedPage ... 135
6.8.10 WebApp.BrowseResources ... 136
6.8.11 WebApp.CreateResource ... 138
6.8.12 WebApp.DeleteResource ... 140
6.8.13 WebApp.RenameResource ... 141
6.8.14 WebApp.DownloadResource ... 142
6.8.15 WebApp.SetResourceVisibility ... 144
6.8.16 WebApp.SetResourceETag ... 145
6.8.17 WebApp.SetResourceMediaType ... 147
6.8.18 WebApp.SetResourceModificationTime ... 148

6.8.19 WebApp.SetVersion ... 149
6.8.20 WebApp.SetUrlRedirectMode ... 151
6.9 Reading and writing process data ... 152
6.9.1 Supported data types ... 152
6.9.2 Parameter assignment of the block properties ... 156
6.9.3 PlcProgram.Read ... 157
6.9.4 PlcProgram.Write ... 159
6.9.5 PlcProgram.DownloadProfilingData ... 162
6.9.6 PlcProgram.Browse ... 167
6.10 Reading and changing the operating mode ... 174
6.10.1 Plc.ReadOperatingMode ... 174
6.10.2 Plc.RequestChangeOperatingMode ... 176
6.10.3 Plc.ReadModeSelectorState ... 177
6.11 Changing time settings via Web API ... 178
6.11.1 Plc.ReadSystemTime ... 178
6.11.2 Plc.SetSystemTime ... 179
6.11.3 Plc.ReadTimeSettings ... 180
6.11.4 Plc.SetTimeSettings ... 183
6.12 Reading diagnostics and service data ... 186
6.12.1 Project.ReadLanguages ... 186
6.12.2 Alarms.Browse ... 189
6.12.3 Alarms.Acknowledge ... 196
6.12.4 DiagnosticBuffer.Browse ... 197
6.12.5 Modules.DownloadServiceData ... 202
6.13 Motion Control ... 204
6.13.1 Technology.BrowseObjects ... 204
6.13.2 Technology.Read ... 206
6.14 Backing up and restoring the configuration ... 207
6.14.1 Plc.CreateBackup ... 207
6.14.2 Plc.RestoreBackup ... 209
6.15 Accessing contents of the SIMATIC load memory ... 212
6.15.1 Files.Browse ... 213
6.15.2 Files.Download ... 216
6.15.3 Files.Create ... 218
6.15.4 Files.Rename ... 219
6.15.5 Files.Delete ... 221
6.15.6 Files.CreateDirectory ... 222
6.15.7 Files.DeleteDirectory ... 224
6.15.8 DataLogs.DownloadAndClear ... 225
6.16 Reading information from SIMATIC Safety ... 226
6.16.1 Failsafe.ReadRuntimeGroups ... 226
6.16.2 Failsafe.ReadParameters ... 228
6.17 Reading CPU information ... 230
6.17.1 Plc.ReadCpuType ... 230
6.17.2 Plc.ReadStationName ... 231
6.17.3 Plc.ReadModuleName ... 231
6.18 Reading information from a redundant system ... 232

6.18.1 Redundancy.ReadSystemInformation ... 232
6.18.2 Redundancy.ReadSystemState ... 235
6.18.3 Redundancy.RequestChangeSystemState ... 236
6.18.4 Redundancy.ReadSyncupProgress ... 237

Glossary ... 240

Index ... 245

Introduction

1

Purpose of the documentation

The web server of SIMATIC S7-1500 CPUs supports three types of web applications:

- System web pages: Standard web pages provided for visualization and access to process data of the CPU
- VoT web pages: Customer-specific web pages developed in the TIA portal using the Virtual of Thing environment
- User-defined web pages: Users can develop their own web pages using JSON APIs

This documentation supports you in setting up and using the web server for system web pages and JSON API for the creation of user-defined web pages.

Basic knowledge required

The following knowledge is required in order to understand the documentation:

- General knowledge in the field of automation technology
- Knowledge of the SIMATIC industrial automation system
- Experience of working with Windows-based computers
- Knowledge about how to use STEP 7 (TIA Portal)
- Knowledge about the use of web servers and web browsers.

Conventions

STEP 7: In this documentation, "STEP 7" is used as a synonym for all versions of the configuring and programming software "STEP 7 (TIA Portal)".

Please also observe notes marked as follows:

NOTE

A note contains important information on the product described in the documentation, on the handling of the product and on the section of the documentation to which particular attention should be paid.

Scope of the documentation

This document is valid for the CPUs:

- SIMATIC S7-1500, ET 200SP and CPU 1513/1516pro as of firmware version V4.0
- SIMATIC S7-1500 Software Controller as of firmware version V40.0
- SIMATIC S7-1500 Virtual Controller as of firmware version V2.0

The documentation contains images of the user interface of the web pages and STEP 7 configuration for a SIMATIC S7-1500 CPU. The images apply as an example for all CPUs.

The images can differ from the interface of the web server in some details, for example depending on the browser used.

NOTE

Web server Function Manual, Edition 11/2023 continues to be valid

With the firmware versions specified above of the CPUs, new "system web pages" have been introduced in contrast to the "Previous web pages". The web pages of previous firmware versions remain valid.

You will find the description of the previous web pages in the Web server Function Manual, Edition 11/2023 (https://support.industry.siemens.com/cs/ww/en/view/59193560), in the "Web pages" chapter.

What's new in the Web Server Function Manual, Version 11/2024 compared to Version 11/2023

What's new?		What are the customer benefits?	Where can I find information?
New contents	Introduction of the system web pages with new design and functionality for: • Overview • Diagnostics • User program • Messages and DataLogs • Maintenance	• Modern design of the user interface • Intuitive and convenient operation • Flexible change management • Interaction with user-defined web applications with reference to Web API sessions	Section System web pages (Page 50)
	New Web API methods	Many new API methods extend your access options to the CPU via the Web API.	An overview of the Web API methods depending on the firmware version of the CPU is available in the section Supported clients (Page 72).
	Introduction of the central user management System-wide, central management of users and user groups outside the TIA Portal via UMC server	User management for extensive automation solutions with several projects, users and user groups Users and user groups can work in all projects in which they are activated and for the appropriate rights have been assigned to them. During operation, users can be added to or removed from a group or their passwords can be changed. All this without the CPU configurations having to be changed and loaded.	Section User management (Page 41)

What's new?		What are the customer benefits?	Where can I find information?
Changed contents	Extension of the scope of this Function Manual to the S7-1500 Virtual Controller	You can apply Web API methods to the CPUs.	An overview of all Web API methods is available in the section Supported clients (Page 72).

Recycling and disposal

For environmentally sustainable recycling and disposal of your old equipment, contact a certified electronic waste disposal service and dispose of the equipment according to the applicable regulations in your country.

Industry Mall

The Industry Mall is the catalog and order system of Siemens AG for automation and drive solutions on the basis of Totally Integrated Automation (TIA) and Totally Integrated Power (TIP).

You can find catalogs for all automation and drive technology products on the Internet (https://mall.industry.siemens.com).

1.1 Function Manuals documentation guide

1.1.1 Information classes Function Manuals

The documentation for the SIMATIC S7-1500 automation system, for the 1513/1516pro-2 PN, SIMATIC Drive Controller CPUs based on SIMATIC S7-1500 and the SIMATIC ET 200MP, ET 200SP, ET 200AL and ET 200eco PN distributed I/O systems is arranged into three areas.

This arrangement enables you to access the specific content you require.

You can download the documentation free of charge from the Internet (https://support.industry.siemens.com/cs/ww/en/view/109742705).

Basic information

The system manuals and Getting Started describe in detail the configuration, installation, wiring and commissioning of the SIMATIC S7-1500, SIMATIC Drive Controller, ET 200MP, ET 200SP, ET 200AL and ET 200eco PN systems. Use the corresponding operating instructions for 1513/1516pro-2 PN CPUs.

The STEP 7 online help supports you in the configuration and programming.

Examples:

- Getting Started S7-1500
- System manuals
- Operating instructions ET 200pro and 1516pro-2 PN CPU
- Online help TIA Portal

Device information

Equipment manuals contain a compact description of the module-specific information, such as properties, wiring diagrams, characteristics and technical specifications.

Examples:

- Equipment manuals for CPUs
- Equipment manuals for interface modules
- Equipment manuals for digital modules
- Equipment manuals for analog modules
- Equipment manuals for communication modules
- Equipment manuals for technology modules
- Equipment manuals for power supply modules
- Equipment manuals for BaseUnits

General information

The function manuals contain detailed descriptions on general topics relating to the SIMATIC Drive Controller and the S7-1500 automation system.

Examples:

- Function Manual Diagnostics
- Function Manual Communication
- Function Manuals Motion Control
- Function Manual Web Server
- Function Manual Cycle and Response Times
- PROFINET Function Manual
- PROFIBUS Function Manual

Product Information

Changes and supplements to the manuals are documented in a Product Information. The Product Information takes precedence over the device and system manuals.

You will find the latest Product Information on the Internet:

- S7-1500/ET 200MP (https://support.industry.siemens.com/cs/de/en/view/68052815)
- SIMATIC Drive Controller (https://support.industry.siemens.com/cs/de/en/view/109772684/en)
- Motion Control (https://support.industry.siemens.com/cs/de/en/view/109794046/en)
- ET 200SP (https://support.industry.siemens.com/cs/de/en/view/73021864)
- ET 200eco PN (https://support.industry.siemens.com/cs/ww/en/view/109765611)

Manual Collections

The Manual Collections contain the complete documentation of the systems put together in one file.

You will find the Manual Collections on the Internet:

- S7-1500/ET 200MP/SIMATIC Drive Controller (https://support.industry.siemens.com/cs/ww/en/view/86140384)
- ET 200SP (https://support.industry.siemens.com/cs/ww/en/view/84133942)
- ET 200AL (https://support.industry.siemens.com/cs/ww/en/view/95242965)
- ET 200eco PN (https://support.industry.siemens.com/cs/ww/en/view/109781058)

1.1.2 Basic tools

Tools

The tools described below support you in all steps: from planning, over commissioning, all the way to analysis of your system.

TIA Selection Tool

The TIA Selection Tool tool supports you in the selection, configuration, and ordering of devices for Totally Integrated Automation (TIA).

As successor of the SIMATIC Selection Tools , the TIA Selection Tool assembles the already known configurators for automation technology into a single tool.

With the TIA Selection Tool , you can generate a complete order list from your product selection or product configuration.

You can find the TIA Selection Tool on the Internet.
(https://support.industry.siemens.com/cs/ww/en/view/109767888)

SIMATIC Automation Tool

You can use the SIMATIC Automation Tool to perform commissioning and maintenance activities on various SIMATIC S7 stations as bulk operations independent of TIA Portal.

The SIMATIC Automation Tool offers a wide range of functions:

- Scanning of a PROFINET/Ethernet system network and identification of all connected CPUs
- Assignment of addresses (IP, subnet, Gateway) and device name (PROFINET device) to a CPU
- Transfer of the date and the programming device/PC time converted to UTC time to the module
- Program download to CPU
- RUN/STOP mode switchover
- CPU localization through LED flashing
- Reading out of CPU error information

- Reading the CPU diagnostic buffer
- Reset to factory settings
- Firmware update of the CPU and connected modules

You can find the SIMATIC Automation Tool on the Internet.
(https://support.industry.siemens.com/cs/ww/en/view/98161300)

PRONETA

SIEMENS PRONETA (PROFINET network analysis) is a commissioning and diagnostic tool for PROFINET networks. PRONETA Basic has two core functions:

- In the network analysis, you get an overview of the PROFINET topology. Compare a real configuration with a reference installation or make simple parameter changes, e.g. to the names and IP addresses of the devices.
- The "IO test" is a simple and rapid test of the wiring and the module configuration of a plant, including documentation of the test results.

You can find SIEMENS PRONETA Basic on the Internet:
(https://support.industry.siemens.com/cs/ww/en/view/67460624)

SIEMENS PRONETA Professional is a licensed product that offers you additional functions. It offers you simple asset management in PROFINET networks and supports operators of automation systems in automatic data collection/acquisition of the components used through various functions:

- The user interface (API) offers an access point to the automation cell to automate the scan functions using MQTT or a command line.
- With PROFIenergy diagnostics, you can quickly detect the current pause mode or the readiness for operation of devices that support PROFIenergy and change these as needed.
- The data record wizard supports PROFINET developers in reading and writing acyclic PROFINET data records quickly and easily without PLC and engineering.

You can find SIEMENS PRONETA Professional on the Internet.
(https://www.siemens.com/proneta-professional)

SINETPLAN

SINETPLAN, the Siemens Network Planner, supports you in planning automation systems and networks based on PROFINET. The tool facilitates professional and predictive dimensioning of your PROFINET installation as early as in the planning stage. In addition, SINETPLAN supports you during network optimization and helps you to exploit network resources optimally and to plan reserves. This helps to prevent problems in commissioning or failures during productive operation even in advance of a planned operation. This increases the availability of the production plant and helps improve operational safety.

The advantages at a glance

- Network optimization thanks to port-specific calculation of the network load
- Increased production availability thanks to online scan and verification of existing systems
- Transparency before commissioning through importing and simulation of existing STEP 7 projects
- Efficiency through securing existing investments in the long term and the optimal use of resources

You can find SINETPLAN on the Internet (https://new.siemens.com/global/en/products/automation/industrial-communication/profinet/sinetplan.html).

1.1.3 SIMATIC Technical Documentation

Additional SIMATIC documents will complete your information. You can find these documents and their use at the following links and QR codes.

The Industry Online Support gives you the option to get information on all topics. Application examples support you in solving your automation tasks.

Overview of the SIMATIC Technical Documentation

Here you will find an overview of the SIMATIC documentation available in Siemens Industry Online Support:

Industry Online Support International (https://support.industry.siemens.com/cs/ww/en/view/109742705)

Watch this short video to find out where you can find the overview directly in Siemens Industry Online Support and how to use Siemens Industry Online Support on your mobile device:

Quick introduction to the technical documentation of automation products per video (https://support.industry.siemens.com/cs/us/en/view/109780491)

YouTube video: Siemens Automation Products - Technical Documentation at a Glance (https://youtu.be/TwLSxxRQQsA)

Retention of the documentation

Retain the documentation for later use.

For documentation provided in digital form:

1. Download the associated documentation after receiving your product and before initial installation/commissioning. Use the following download options:
   - Industry Online Support International: (https://support.industry.siemens.com)

     The article number is used to assign the documentation to the product. The article number is specified on the product and on the packaging label. Products with new, non-compatible functions are provided with a new article number and documentation.
   - ID link:

     Your product may have an ID link. The ID link is a QR code with a frame and a black frame corner at the bottom right. The ID link takes you to the digital nameplate of your product. Scan the QR code on the product or on the packaging label with a smartphone camera, barcode scanner, or reader app. Call up the ID link.
2. Retain this version of the documentation.

Updating the documentation

The documentation of the product is updated in digital form. In particular in the case of function extensions, the new performance features are provided in an updated version.

1. Download the current version as described above via the Industry Online Support or the ID link.
2. Also retain this version of the documentation.

mySupport

With "mySupport" you can get the most out of your Industry Online Support.

Registration	You must register once to use the full functionality of "mySupport". After registration, you can create filters, favorites and tabs in your personal workspace.
Support requests	Your data is already filled out in support requests, and you can get an overview of your current requests at any time.
Documentation	In the Documentation area you can build your personal library.
Favorites	You can use the "Add to mySupport favorites" to flag especially interesting or frequently needed content. Under "Favorites", you will find a list of your flagged entries.
Recently viewed articles	The most recently viewed pages in mySupport are available under "Recently viewed articles".
CAx data	The CAx data area gives you access to the latest product data for your CAx or CAe system. You configure your own download package with a few clicks: • Product images, 2D dimension drawings, 3D models, internal circuit diagrams, EPLAN macro files • Manuals, characteristics, operating manuals, certificates • Product master data

You can find "mySupport" on the Internet. (https://support.industry.siemens.com/My/ww/en)

Application examples

The application examples support you with various tools and examples for solving your automation tasks. Solutions are shown in interplay with multiple components in the system - separated from the focus on individual products.

You can find the application examples on the Internet.
(https://support.industry.siemens.com/cs/ww/en/ps/ae)

Safety instructions

2

2.1 General security information

Note the security-relevant information provided in the corresponding system manual.

You can find information relevant to cybersecurity in the section Industrial cybersecurity (Page 17).

Industrial cybersecurity

3

Due to the digitalization and increasing networking of machines and industrial plants, the risk of cyber attacks is also growing. Appropriate protective measures are therefore mandatory, particularly in the case of critical infrastructure facilities.

Refer to the System Manual and in the Security guideline for SIMATIC HMI operator devices and SIMATIC WinCC Unified (https://support.industry.siemens.com/cs/ww/de/view/109481300/en) for general information and measures regarding industrial cybersecurity.

This section provides an overview of security-related information pertaining to the communication of your SIMATIC system.

NOTE

Security-relevant changes to software or devices are documented in the section Introduction (Page 7).

3.1 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
https://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://new.siemens.com/cert.

3.2 Cybersecurity-relevant information in this manual

Observe all security-relevant notes on topics in this Function Manual.

Security-relevant notes on ...	**Section**
Activation of the Web server for specific interfaces	Configuring the web server (Page 22)
Secure communication: Secure data traffic through access via secure "HTTPS" transfer protocol Usage of CA-signed or self-signed web server certificates	Configuring the web server (Page 22) Certificates (Page 29)
Setting up local or central user management	User management (Page 41)
Changing the password in runtime via Web API	Api.ChangePassword (Page 91)

Setting up and calling a web server

4

4.1 Properties of the Web server

Benefits of the web server

The web server enables monitoring and administering of the CPU by authorized users over a network. Evaluations, diagnostics, and modifications are thus possible over long distances. Monitoring and evaluation is possible without STEP 7, only a web browser is required. Note that you must take appropriate measures to protect the CPU from compromise (such as restricting network access, using firewalls).

Activating the web server

The web server is deactivated in the delivery state of the CPU. This means that you must load a project in which the web server is activated to enable access.

Security functions

The web server provides the following safety functions:

- Access via the secure "HTTPS" transfer protocol using the CA-signed or self-signed web server certificate
- Authentication via local or central user management
- Activation for specific interfaces
- Changing the password of local users in runtime via Web API

Web browser

You require a web browser to access the web pages of the CPU. You can use the web browsers listed below to access the web server. The web pages are fundamentally based on new functions that are supported by new web browsers.

Table 4-1 Supported web browsers

Web browser*	Operating system
Microsoft Edge	Windows 10
Mozilla Firefox	Windows 10
Google Chrome	Windows 10

* Not included in the scope of delivery of the product described here

Web browser*	Operating system
Chrome Mobile	Android
Safari Mobile	iOS
HMI BrowserControl	HMI Unified Comfort Panels

* Not included in the scope of delivery of the product described here

Always keep the operating system and web browsers up-to-date and protect your computer on the basis of the latest recommendations for IT security.

NOTE

Two reserved communication connections are available to the web server for communication with the CPU.

Depending on the web browser used, a different number of connections to the CPU are established. If more connections are available, more communication connections will be established.

If no further connections are available, display or functional problems may occur. This is because the web server can reject all further communication connections apart from the two that are reserved if resources problems arise. For this reason, the web pages may not load fully.

For more details about connections and connection resources, refer to the Communication Function Manual (https://support.industry.siemens.com/cs/ww/en/view/59192925).

NOTE

After a firmware update of the CPU, incorrect display of web pages can occur in various web browsers.

Solution: Press F5 or clear the web browser cache.

NOTE

Web browser behavior can be different if a certificate is not valid yet or is no longer valid. Siemens has no influence on this behavior. Usage of self-signed certificates can also cause problems.

This means we cannot guarantee the reliable functionality of the web server with self-signed and invalid certificates.

System web pages as of the current firmware version of the CPU

As of the current firmware version of the CPUs S7-1500, a part of the web pages are available in a new, modern design. This makes solutions on the basis of modern web technologies possible.

The new web pages are supplied by default with the firmware of the CPU and are called "system web pages".

The system web pages can be used as of the following configured firmware version of the CPUs:

- CPUs S7-1500, ET 200SP or CPU 1513/1516pro as of firmware version V4.0 (except for R/H-CPUs)
- CPUs S7-1500 Software Controller as of firmware version V40.0
- CPUs S7-1500 Virtual Controller as of firmware version V2.0

More information

The system web pages are described in the section System web pages (Page 50).

For information on calling up the system and previous web pages refer to the section System web pages and previous web pages (Page 26).

Web access to the CPU via PG/PC, HMI devices and mobile end devices

Proceed as follows to access the web server:

1. Use STEP 7 to download a project in which the web server is activated to the CPU.
2. Connect the display device (PG/PC, HMI, mobile terminal device) with the CPU or a communications module using a PROFINET interface.
   If you are working with WLAN, activate the WLAN on the display device and establish a connection to the access point (e.g. SCALANCE W788-1RR or SCALANCE W784-1), which is in turn connected to the CPU.
3. Open the web browser on the display device.
4. Enter the IP address of the interface of the CPU which is connected to the client in the "Address" field of the web browser in the following format: https://a.b.c.d (entry example: https://192.168.3.141). The login page of the system web pages opens.

More information

More information on access using the secure "HTTPS" transfer protocol is available in the section Configuring the Web server (Page 22).

You can access the web server of the CPU via WLAN with the smartphone. You can find more information in the FAQ entry ID 103473392 (https://support.industry.siemens.com/cs/ww/en/view/103473392) on the Internet.

4.2 Configuring the Web server

To use the full functionality of the Web server, the following settings in STEP 7 are necessary.

Activate Web server on this module

The Web server is deactivated in the default setting of a configured CPU. Proceed as follows to activate the Web server:

1. Open the "Devices & Networks" view by double-clicking in the project tree in STEP 7.
2. Select the desired CPU in the device, network or topology view.
3. Navigate to the "Web server" area in the inspector window properties, "General" tab.
4. Activate the "Activate web server on this module" check box.
   In the process the following message is output:

Figure 4-1 Security note upon activation of the Web server in STEP 7

NOTE

When projects from deliveries are applied in which the Web server was already activated and configured on the module, this security note is not shown.

NOTE

Activating Web server for R/H-CPUs

In the case of R/H-CPUs you can only activate the Web server for both R/H-CPUs and with the identical settings. In addition to the Web server properties, this also applies to the "Overview of the interfaces", for the certificates and the user management.

When both CPUs have loaded the project, you can use the IP addresses of both CPUs to use the Web server.

Depending on the CPU used, you can make your own settings or the settings are fixed.

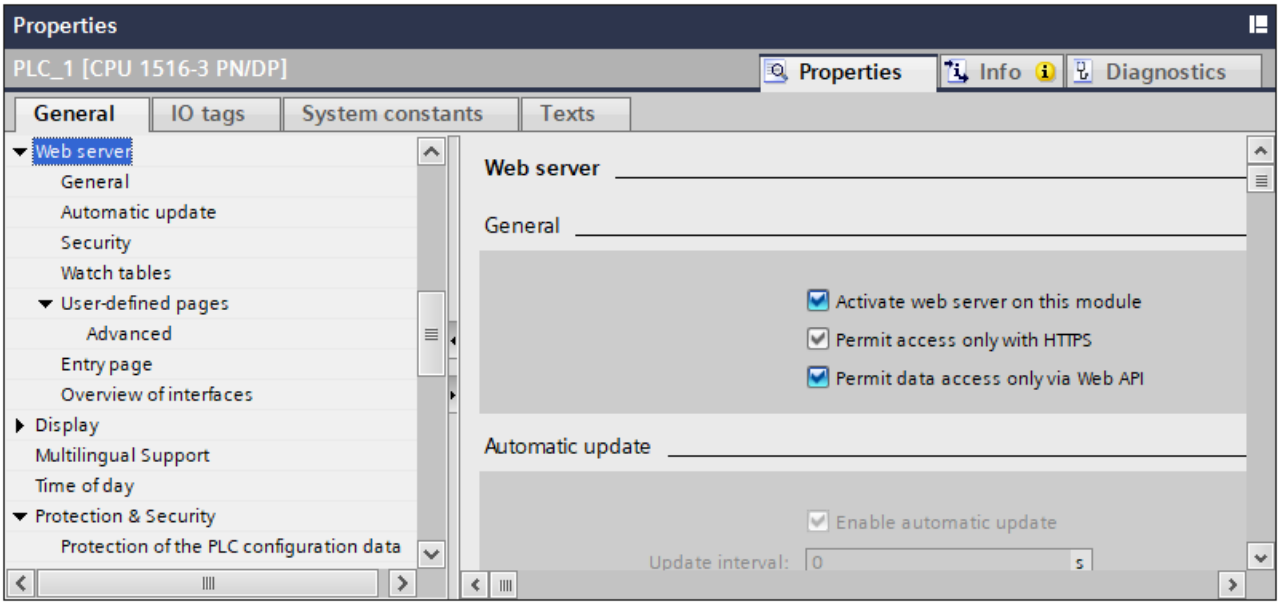


Figure 4-2 Web server settings in STEP 7

Permit access only with HTTPS

The secure transfer protocol "HTTPS" is always activated in the web server (via Port 443) and cannot be deactivated. Prerequisite for operating the web server with HTTPS is a valid web server certificate. You can find information on how to create and assign web server certificates in the section Managing certificates via TIA Portal (Page 30).

To ensure secure access to the Web server the "Permit access only with HTTPS" check box is activated in the basic setting of a configured CPU. In the case of R/H-CPUs, data access is only possible via HTTPS and is therefore preset and cannot be changed.

By default, data exchange between client and web server takes place via a secure connection and is therefore protected against attacks by third parties. Note that the URL of the CPU starts with "https://" in this case.

The requirements for error-free HTTPS access to the CPU are as follows:

- The current date/time must be set in the CPU.

  NOTE

  When using secure communication (e.g. HTTPS), make sure that the corresponding modules have the current time of day and the current date.

- The IP addresses of the CPU must be assigned.
- A valid certificate from a certificate authority is installed in the web browser, on which the web server certificate in the CPU is based.

Permit data access only via Web API

If you select the "Permit data access only via Web API" check box, only Web API-based functions are available, including:

- JSON-RPC interface
- Ticketing and web applications including system web pages

Functions on the Web server via the unencrypted HTTP protocol and AWP commands are no longer accessible.

If you select the "Permit data access only via Web API" check box, the "Permit access only with HTTPS" check box is automatically selected and cannot be changed.

For R/H-CPUs, data access is only possible via Web API and is therefore preset and cannot be changed.

The setting of the "Permit data access only via Web API" check box influences whether you can access the previous web pages. You can find more information in section System web pages and previous web pages (Page 26).

Security: Certificates, access protection and user management

STEP 7 automatically generates and uses a valid web server certificate for the CPUs.

The encrypted connection created with the help of the certificate prevents eavesdropping or falsification of communication, but does not provide access protection. This means you have to protect your CPU from unauthorized access with the corresponding configuration in the user management.

You can find information on how to create and manage certificates in the section Certificates (Page 29).

The procedure for setting up the user management with password-protected users for the Web server is based on the configured firmware version of your project. You can find more information in section User management (Page 41).

You can find more information on access protection on the CPU in the STEP 7 online help, keyword: "Protection".

Entry page

The setting in the "Entry page" section is only relevant if you use the previous web pages and the previous user program for the CPUs. In this case, you can select a start page for the web server in the "Entry page" section. You can find more information on calling the previous web pages in the section System web pages and previous web pages (Page 26).

Activation of the Web server for specific interfaces

In the "Overview of interfaces" area, you have the option to enable access to the Web server.

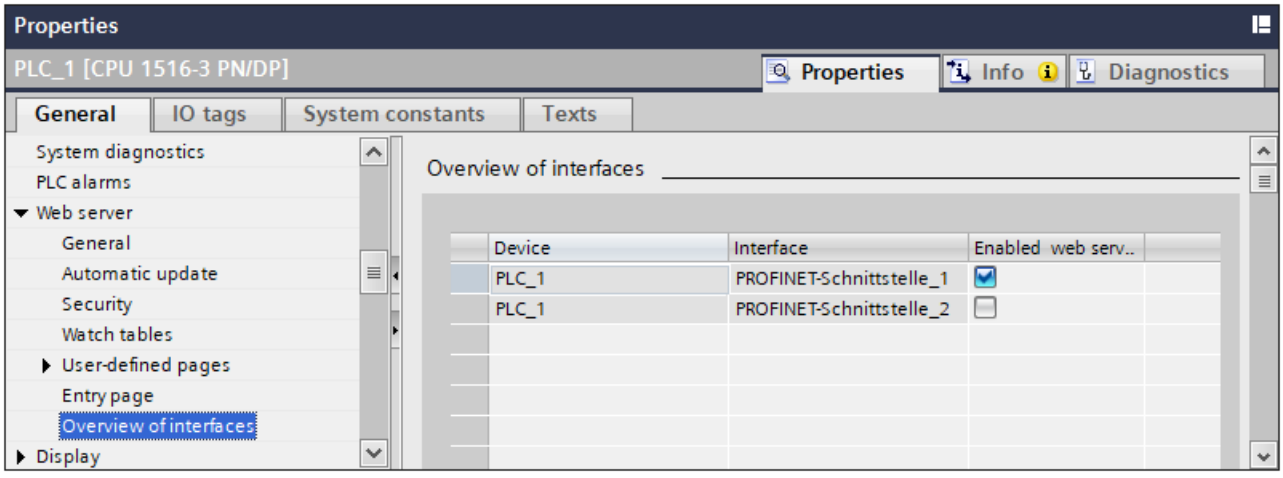


Figure 4-3 Enabling access to the Web server via the interfaces

Setting the project languages for the Web server

In total, you can assign up to three different project languages to the user interface languages of the Web server.

In STEP 7, activate the project languages that you want to use and then assign one of the activated project languages to each of the Web server interface languages.

You can find more information about the language settings and a description of how to assign a project language to the interface languages in the section Language settings (Page 39).

NOTE

The language can only be changed in STOP mode. Subsequently a download is carried out.

4.3 System web pages and previous web pages

As of the firmware version ≥ V4.0/40.0/2.0 of the CPUs, a part of the web pages are available in a new, modern design. This makes solutions on the basis of modern web technologies possible. The new web pages are characterized by their intuitive and convenient operation.

The new web pages are supplied by default with the firmware of the CPU and are called "system web pages". Through the system web pages, interaction with user-defined web applications in the Web API sessions becomes possible.

In future you can update the system web pages independently of the CPU or, if required, retain an older version.

The web pages of the CPUs with a firmware version < V4.0/40.0/2.0 are called "previous web pages".

Extended functionality of the web server:

The web server functionality has been extended as follows with firmware version V4.0/V40.0/V2.0:

- The existing web application end point makes the contents of the system web pages available.
- You activate or deactivate the system web pages in STEP 7.
- You specify a system or previous web page as the web server default page in STEP 7.

Overview: Displaying system web pages and previous web pages

Which web pages are displayed depends on the configured firmware version of the CPU and on the setting in STEP 7 for "Permit data access only via Web API" (see section Configuring the web server (Page 22)).

Table 4-2 Web page display and firmware version of the CPU

Configured firmware version of the CPU			Available web pages	Default pages of the web server	Where can I find information?
S7-1500/ ET 200SP/ ET 200pro	Software controller	S7-1500V			
≤ V3.1[1]	≤ V30.1	V1.0	Previous web pages	Previous web pages	Web server Function Manual (https://support.industry.siemens.com/cs/ww/en/view/591-93560), Edition 11/2023
≥ V4.0[1]	≥ V40.0	≥ V2.0[2]	With "Permit data access only via Web API" check box selected: System web pages	System web pages	Present section
			With "Permit data access only via Web API" check box cleared: Previous and system web pages in parallel[2]	Previous web pages[2] You can configure the system web pages as default pages via the project tree.	Present section

1 S7-1500R/H-CPUs do not support web pages. This means that the web server only functions via the Web API.

2 S7-1500V-CPUs support only the system web pages. Therefore these are displayed.

Configuring system web pages in STEP 7

If you have set "Permit data access only via Web API" in the hardware configuration of the CPU, only the system web pages and the Web API are available.

You use STEP 7 for the following tasks:

- To deactivate/deactivate the system web pages
- To configure any URL of a web application as the default page of the web server

The system web pages are a special type of a web application. In the project tree in STEP 7, you select the "General settings" menu under the CPU in the "Web applications" folder.

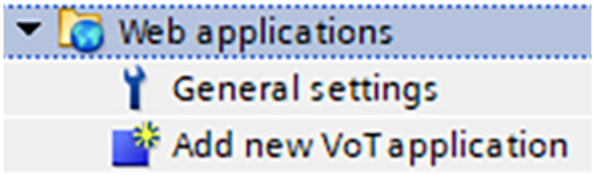


Figure 4-4 Project navigation web applications

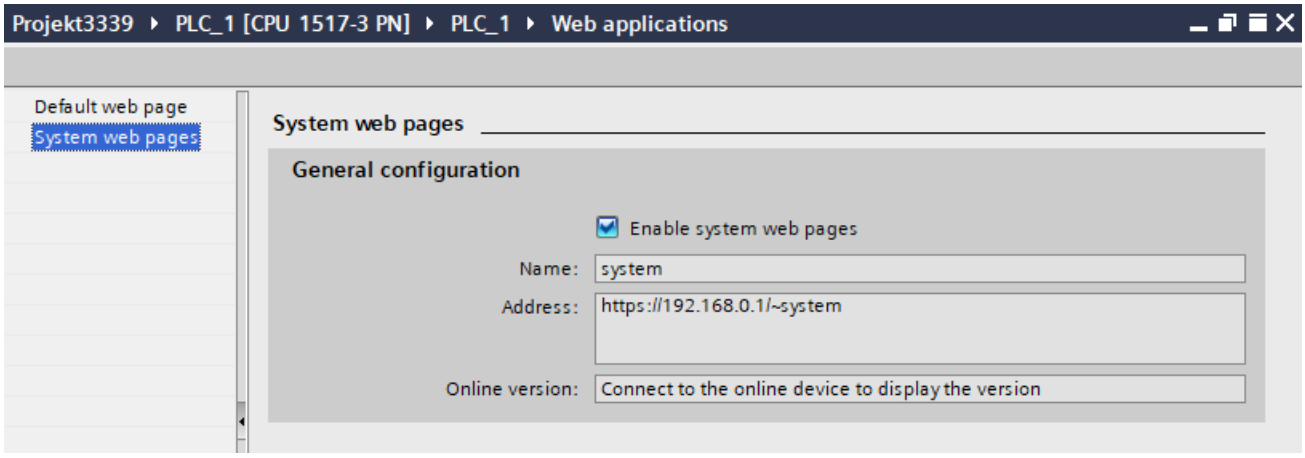


Figure 4-5 Project navigation general settings web server

For a configured CPU as of firmware version V4.0/V40.0/V2.0, you can deactivate the system web pages here or activate them again. You can change the setting later via the Web API method WebServer.SetDefaultPage (Page 115), independently of the operating mode of the CPU.

Configuring the web server default page in STEP 7

You can configure the default page of the web server in the "General settings" menu.

1. Select "Default web page".
2. Select the web application whose overview you want to configure as the default page:
   - The system web pages
   - The web server entry page specified in the hardware configuration
   - A VoT application

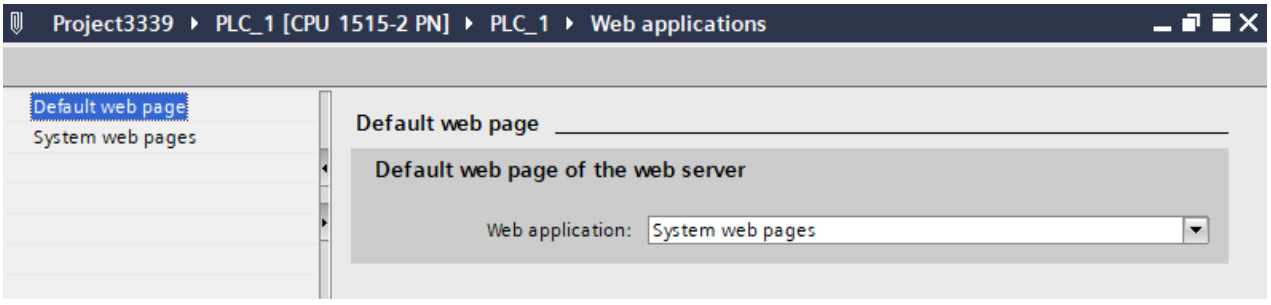


Figure 4-6 Project navigation default page web server

Connecting to the Web server

You establish a connection to the Web server by entering the IP address of the interface of the configured CPU which is connected to the client in the address bar of the web browser.

The URL address differs depending on whether you want to call up the system web pages or the previous web pages:

- System web pages, for example via: https://192.168.3.141/~system
- Previous web pages, via: https://182.168.3.141/Portal/Portal.mwsl?PriNav=Start

The connection is established and the login page or overview of the system web pages (intro page of the web server at previous sites) is opened.

More information

The section System web pages (Page 50) contains the structure and describes the contents of the system web pages.

The description of the previous web pages is contained in section 4 of the Web server Function Manual (https://support.industry.siemens.com/cs/ww/en/view/59193560), Edition 11/2023

4.4 Certificates

4.4.1 Web server certificates

To secure the data exchange with a partner against interception of the communication, different applications and communication functions of the CPU use device certificates that are managed application-specifically. In the case of the device certificate for the web server, this involves a web server certificate.

The new system web pages, user pages on the basis of the web applications as well as the Web API are only available via HTTPS. Operation of the Web server using the secure transfer protocol HTTPS requires a valid Web server certificate.

The previous web pages and user-defined pages can also use the unencrypted transfer protocol.

You manage certificates in a number of ways.

4.4.2 Managing certificates via TIA Portal

Creating and assigning a Web server certificate

The secure transfer protocol "HTTPS" is always activated in the web server (via Port 443) and cannot be deactivated. Prerequisite for operating the web server with "HTTPS" is a valid web server certificate.

The previous web pages also allow the unencrypted transfer protocol "HTTP". The new system web pages support only "HTTPS".

Configuring the web server for individual protocols is described in the section Configuring the Web server (Page 22).

You create the web server certificate of the CPU using STEP 7 and assign it to the web server in the properties of the CPU. This certificate is also downloaded to the CPU automatically when the hardware configuration is downloaded.

You can create different Web server certificates:

- If you use the global security settings for the certificate manager, the certification authority of the project (CA certificate) signs the device certificate of the Web server. During loading, the **CA certificate** of the project is automatically loaded as well.
- If you do not use the certificate manager in the global security settings, STEP 7 generates the device certificate as a **self-signed certificate**.

NOTICE
Security-relevant functions only possible with CA-signed/self-signed web server certificate The security-relevant functions of backup and restoring the CPU configuration, see section Online backup, are only possible with a CA-signed or self-signed web server certificate. A valid CA-signed/self-signed web server certificate in the CPU is furthermore required for user authentication with password-protected users. Recommendation: To use the full functionality of the Web server, use the Certificate Manager to create a CA-signed or self-signed web server certificate in the global security settings and assign it to the CPU.

If no certificate is set up in the web browser, a warning appears recommending that you do not use the page. To view the page, you may need to "Add an exception", depending on the web browser used.

You can find instructions for installing the certificate in the help system of your web browser and in the FAQ with the entry ID 103528224 (https://support.industry.siemens.com/cs/ww/en/view/103528224) on the Service&Support web page.

NOTE

To protect against manipulation from the outside, download the certificate only in an environment that is guaranteed not to be compromised. Installation of the certificate has to be carried out once for each display device you wish to use.

Advantages and disadvantages of self-signed certificates

The self-signed certificates offer the following advantages:

- No fees for the creation of the signature
- Rapid installation. You do not require a response from the certificate authority

The self-signed certificates have the following disadvantages:

- Threat to personal user data
- Permanent warning "Unknown publisher" in the web browser
- Data security is not ensured
- Lack of user trust because a signature of the certificate authority is missing

The self-signed certificates are subject to the following restrictions:

- Self-signed certificates are not embedded in a "Public Key Infrastructure" (PKI) and cannot sign other certificates.
- The certificate only applies to the configured CPU and is not available throughout the project.
- Private keys of the certificate cannot be exported.

NOTE

Recommendation: Use the CA-signed web server certificates to use the functions of the CPU optimally.

Creating and assigning a CA-signed Web server certificate

To create a CA-signed Web server certificate with TIA Portal, follow these steps:

1. In the project tree, navigate to "Security settings > Settings".
2. Click in the "Project protection" section on the "Protect this project" button and assign a user name and password for the project administrator. A new "Certificate manager" entry is displayed under "Security functions" in the project tree.

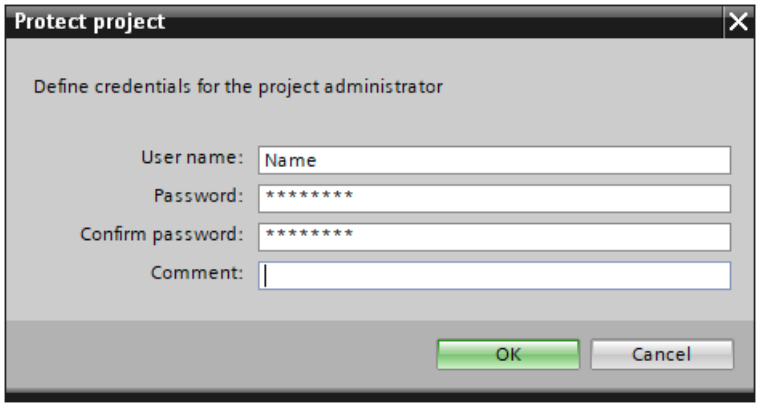


Figure 4-7 Protecting a project

3. In the "General" tab of the Properties of the CPU Inspector window, navigate to the "Protection & Security > Certificate Manager" area and select the "Use global security settings for certificate manager" option.

   NOTE

   For managing certificates with the global security settings, you require the "Configure security" configuration permission.

4. Log in as a user in the project tree in the "Security settings" section. For a new project, the "Administrator" role is intended for the first login.
5. In the Inspector window Properties of the CPU, "General" tab, navigate to the "Web server > Security" area.
6. Click the "..." button in the drop-down list to select a certificate.
   The dialog for selecting and creating a new certificate opens.

7. Click the "Create" button in the drop-down list to select a certificate.
   The "Create certificate" dialog opens.

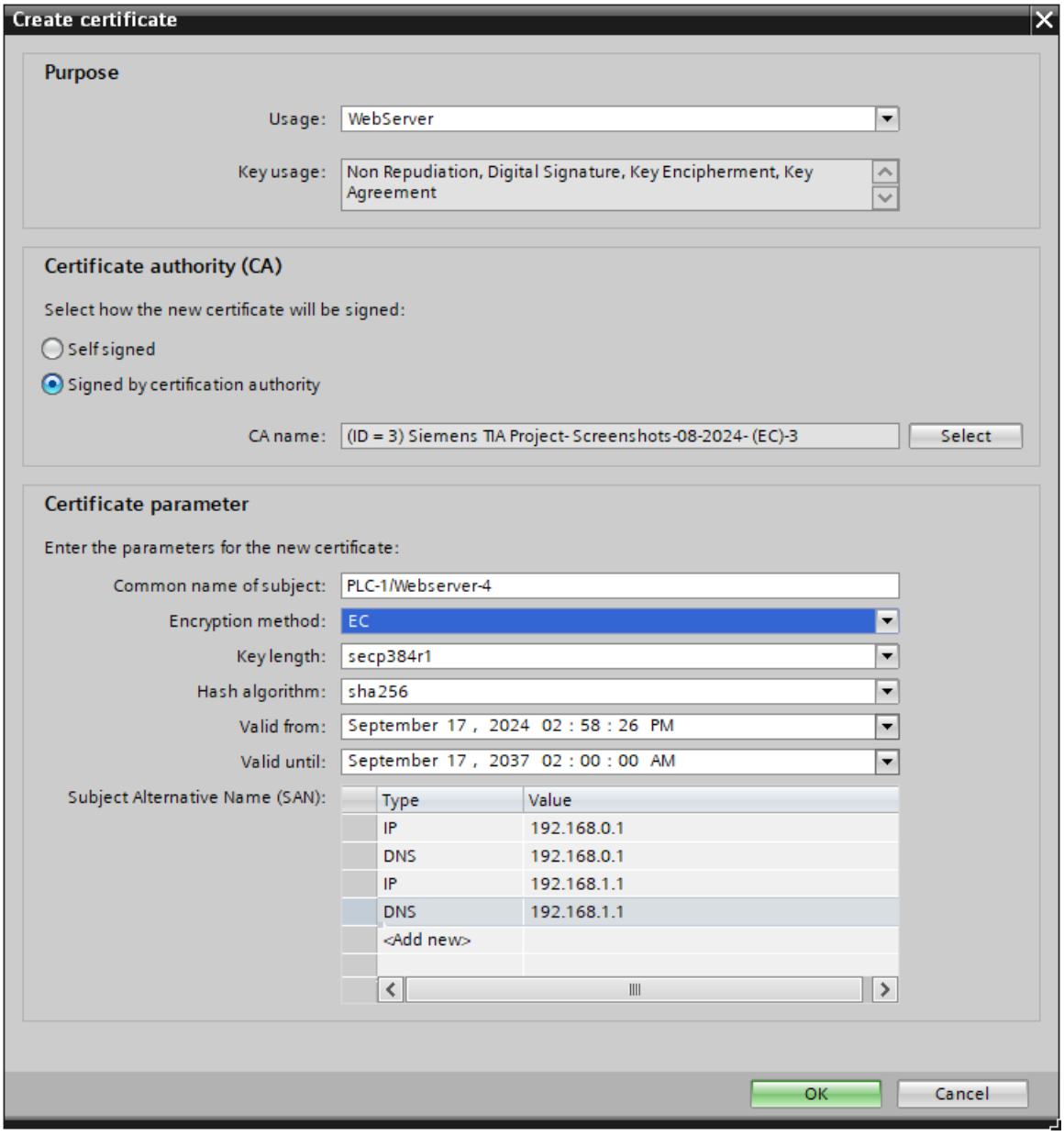


Figure 4-8 Creating a CA certificate

8. In the follow-up dialog box, select the "Signed by certificate authority" check box and select the certificate authority from the drop-down list.

9. Enter the parameters for the new certificate or confirm the default settings.
   - Select "Web server" in the "Usage" box.
   - RSA as well as EC certificates are supported by the web server. The advantage of the EC certificates is the higher performance provided.
   - Enter the IP address(es) of the interface(s) or the domain name of the configured CPU in the "Subject Alternative Name" field.
10. Click "OK" to confirm.
11. Compile and load the configuration in the CPU.
    The device certificate of the Web server and the CA certificate are components of the configuration.

NOTICE
Addressing the Web server of the CPU via domain names If you enter the IP address(es) of the interface(s) of the configured CPU in the "Subject Alternative Name" field, the generated certificate may not be accepted by all Internet browsers. In addition, you must generate and load a new Web server certificate (end entity certificate) with each change of the IP address of an Ethernet interface of the CPU, since the identity of the CPU changes with the IP address. You can avoid this problem by addressing the Web server of the CPU using domain names instead of IP address(es), e.g. "myconveyer-cpu.room13.myfactory.com". For this purpose, you have to manage the domain names of your CPU via a DNS server. Addressing via domain names is recommended especially for a configuration with reception of the IP address from a DHCP server, as in this case the assigned IP address is not known beforehand.

Creating a self-signed Web server certificate

To create a self-signed Web server certificate with TIA Portal, follow these steps:

1. In the Inspector window Properties of the CPU, "General" tab, navigate to the "Web server > Security" area.
2. Click the "..." button in the drop-down list to select a certificate.
   The dialog for selecting and creating a new certificate opens.
3. To select an existing certificate, double-click a certificate of the list.
4. To create a new certificate, click the "Create" button.
   The "Create certificate" dialog opens.

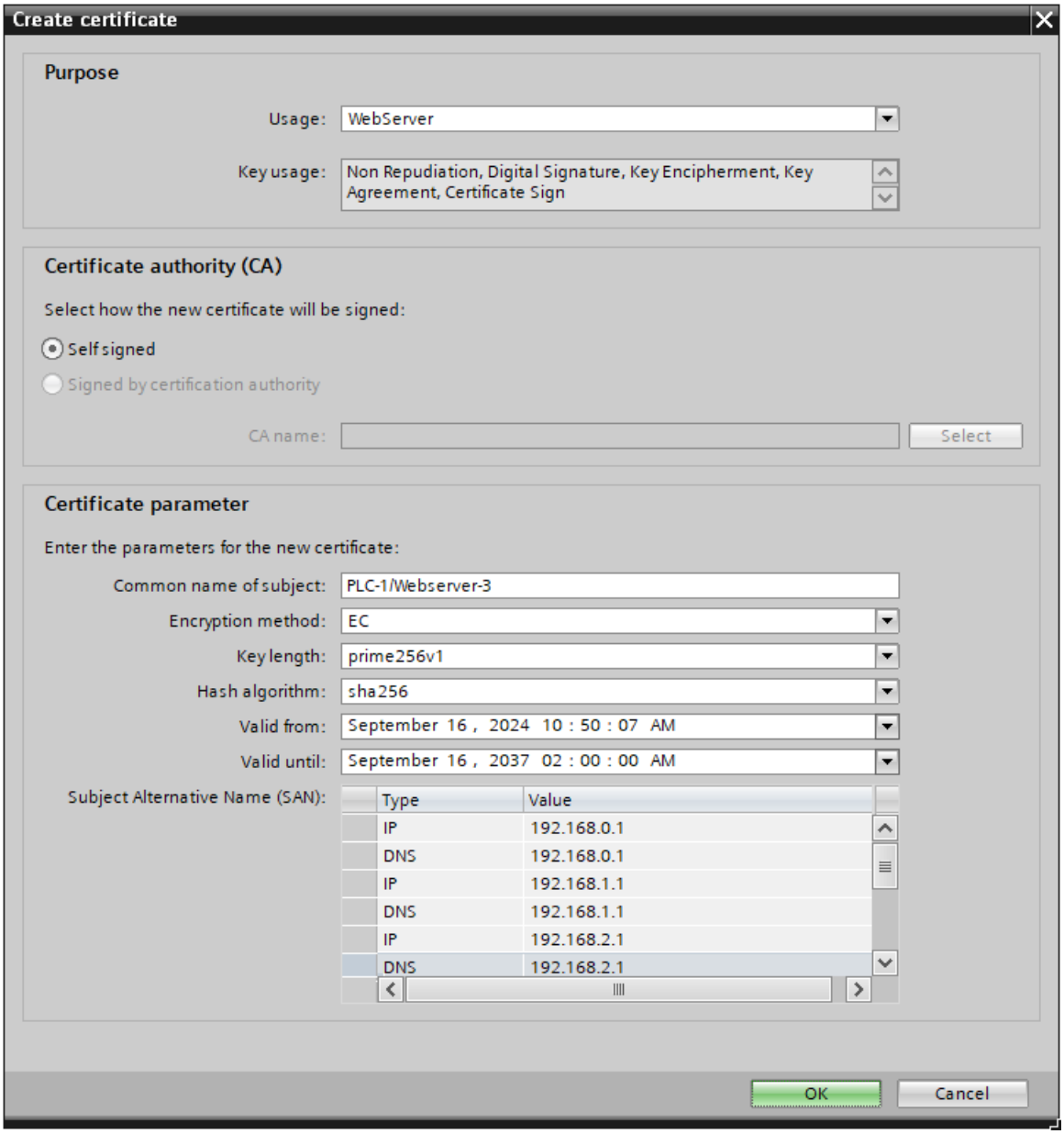


Figure 4-9 Creating a self-signed certificate

5. Select the "Self-signed" check box.
6. Enter the parameters for the new certificate or confirm the default settings.
   - Select "Web server" in the "Usage" box.
   - RSA as well as EC certificates are supported by the web server. The advantage of the EC certificates is the higher performance provided.
   - Enter the IP address(es) of the interface(s) or the domain name of the configured CPU in the "Subject Alternative Name" field.
7. Click "OK" to confirm.
8. Compile and load the configuration into the CPU.
   The device certificate of the Web server is a component of the configuration.

More information

For detailed information on local self-signed and global CA-signed certificates, on the "Public Key Infrastructure" (PKI) and on certificate management, refer to the Communications Function Manual (https://support.industry.siemens.com/cs/ww/en/view/59192925) and to the STEP 7 online help, keyword "Secure communication".

The application example "Using certificates with TIA Portal (https://support.industry.siemens.com/cs/ww/en/view/109769068)" includes detailed instructions on how to create a secure connection to the web server of a SIMATIC S7-1500 CPU.

4.4.3 Managing certificates in runtime

If you manage certificates via the TIA Portal, load a certificate together with the hardware configuration into the CPU. To do this, the CPU must be in STOP mode. You cannot load a new certificate or renew an existing certificate without a RUN-STOP-RUN transition. To this purpose the global certificate management must be used and a certificate renewed.

If you manage certificates at runtime of the CPU, loading or updating a certificate is also possible in RUN mode.

Managing the web server certificate during the CPU runtime

As of firmware version V3.0, it is also possible to transfer web server certificates to the CPU during runtime via the GDS server using OPC UA methods. The GDS server is part of the OPC UA server in the CPU. Through GDS push management functions, you can automatically update OPC UA certificates for the OPC UA server of the S7-1500 CPU.

As of firmware version 3.1, the certificate can be exchanged during runtime. No system STOP is required - the system continues to run.

You can find detailed information about the concept of automated certificate management with GDS (Global Discovery Services) in the Communication Function Manual (https://support.industry.siemens.com/cs/ww/en/view/59192925).

Setting the type of certificate management

In the "Protection & Security" > "Certificate manager" category on the "General" tab of the "Properties" Inspector window, select how you want to handle certificates.

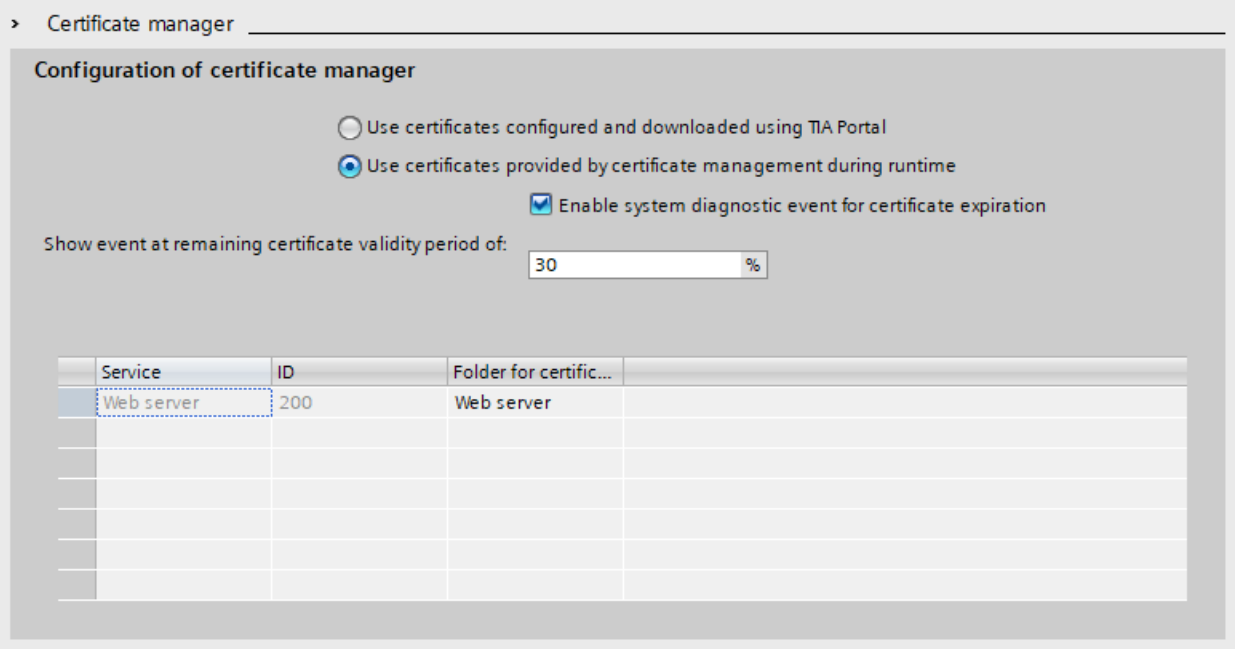


Figure 4-10 Configuration of the certificate manager

If you want to submit certificates via GDS at runtime, click the "Use certificates provided by certificate management during runtime" option.

By selecting the "Enable system diagnostic event for certificate expiration" button, you specify that you want to be notified when a certificate expires. In the input field "Show event at remaining certificate validity period of:" enter a percent value. At the time this value is reached, the CPU triggers a system alarm with a maintenance request.

Example:

The certificate transferred via GDS on 2024-06-01 is valid from 2024-06-01 to 2024-06-30 (30 days). You have input a percent value of 10 for the diagnostics event. On 2024-06-27, after 90% of the validity period has expired, the system diagnostics alarm reports that the transmitted certificate will expire on 2024-06-30.

Regardless of the configured percentage value, a message appears in any case when the validity period of a certificate expires.

In the lower area of the "Certificate manager" category in the table, you can find a list of all CPU applications with certificates you may transfer to the CPU at runtime. In the list, the CPU applications are assigned an ID. Under the "Folder for certificate repository at runtime" column, you can find the changeable name of the certificate group.

Handling of existing certificates during loading

Before you load a project into the CPU, you can determine in the "Load preview" dialog window what should happen with the certificates of the CPU received during runtime.

As of firmware version V3.0, you can use the "Delete selected" option to delete certificates of selected CPU applications.

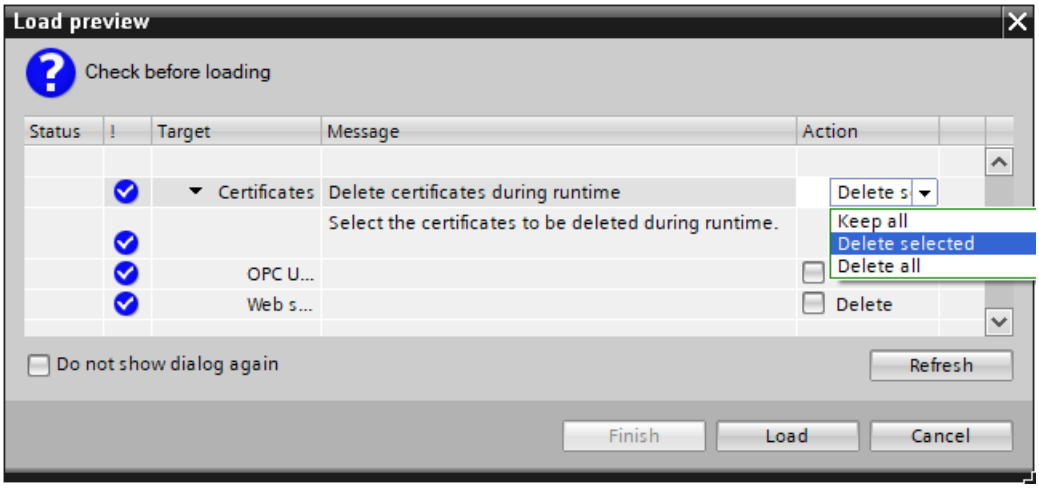


Figure 4-11 Deleting certificates

4.4.4 Installing certificates for the web browser

In order to use the web browsers for HTTPS pages, you have to install the correct certificates in the web browser. The installation is described below for the web browsers "Firefox", "Chrome", and "MS Edge":

Installing certificates in Firefox

1. Ensure that the certificate that you want to import into Firefox is stored on your computer.
2. Start "Firefox" and then click at the top right on the button to open the menu.
3. Select the "Settings" entry.
4. Then change to the left to the "Data protection & security" menu and scroll completely downwards on this page.
5. The "Certificates" section is located here. Click the "Display certificates" button.
6. Click at the bottom on "Import..." and navigate to your certificate file.
7. Click "OK".

The entry is included subsequently in Firefox. Should there be any problems, ensure that the most recent version of Firefox is installed.

Installing certificates in Chrome or MS Edge

1. Ensure that the certificate that you want to import into "Chrome" or "MS Edge" is stored on your computer.
2. Open the settings or go to chrome://settings.
3. Click "Data protection and security" on the left.
4. Click "Security".
5. Scroll to "Advanced".
6. Click "Manage certificates".
7. Click "Import".
8. Click "Next" and navigate to the storage location of the downloaded certificate. Select it and click "Open".
9. Click "Next".
10. Click "Store all certificates in the following memory".
    - If you want to use the certificate generated by the server, select "Trusted root certification authorities".
    - If you want to use a certificate from a different source, select "Intermediate certification authorities".
11. Click "Next" and then "Finish".
12. Click "OK".

Installing certificates in Safari

A detailed description is available in the respective documentation for iOS- or Android-based products.

4.5 Language settings

Introduction

The Web server provides the user interface in the following languages: The defaults is English (USA):

- German
- English (USA)
- French
- Italian
- Spanish
- Japanese
- Chinese (Simplified)

- Korean
- Russian
- Turkish
- Portuguese (Brazil)

You can set the user interface language in the header of the overview of the new system web pages, see section Overview (Page 58).

Requirement for multilingual output of texts

In order for the Web server to correctly display messages, comments, and diagnostics information in the different project languages, you must assign a project language to each of the desired user interface languages of the Web server in STEP 7.

NOTE

The project languages of the STEP 7 project that you want to assign must be activated and the corresponding texts (translations) must be available in the project. You can find a list of available project languages in the project tree under "Languages & resources".

Configuring a project language for the Web server

Once you have activated the Web server on your module, assign an active project language from the drop-down list to each interface language. This assignment affects the display of pages with project language-dependent texts, such as the diagnostics buffer.

1. Navigate to the "Multilingual support" area in the "General" tab of the CPU properties in the inspector window.
2. Assign an active project language from the drop-down list to each interface language.

Overall, you can assign up to three different project languages to the user interface languages of the web server.

NOTE

A project language can only be added in STOP mode. Subsequently a download is carried out.

Reference

You can find more information on how to set the project language in STEP 7 in the STEP 7 online help, keyword: "Selecting project languages".

4.6 User management

User authentication on the Web server always takes place via secure HTTPS communication. You can authenticate yourself via the system web pages or using the Web API.

The CPU supports various possibilities for the user management.

Possibilities - setting up user management

You set up the user management in TIA Portal depending on the configured and loaded firmware version of the CPU.

Table 4-3 User management and firmware version of the CPU

Configured firmware version of the CPU			User management			Where can I find information?
S7-1500/ ET 200SP/ ET 200pro	Software controller	S7-1500V	Static	Local	Central	
≤ V3.0	≤ V30.0	-	x	-	-	Web server Function Manual (https://support.industry.siemens.com/cs/ww/en/view/5919-3560), Edition 11/2023
V3.1	V30.1	V1.0	-	x	-	Present section
≥ V4.0	≥ V40.0	≥ V2.0	-	x	x	Present section, S7-1500, ET 200MP System Manual (https://support.industry.siemens.com/cs/ww/en/view/5919-1792)

Static user management: You can create users, define access rights and assign passwords via the user list. You only have access to the options that are permanently linked to the access rights.

Local user management: You manage all project users along with their rights (for example, function rights) for all CPUs in the project in the editor for users and roles of the project in TIA Portal.

Central user management: The central user management allows system-wide, central management of users outside TIA Portal. The users and user groups can work in all projects in which they are activated and for the appropriate rights have been assigned to them.

Local user management

As of TIA Portal version V19 and CPU firmware version V3.1/V30.1/V1.0, the CPUs dispose of an improved management of users, roles, and CPU function rights (User Management & Access Control, UMAC).

You manage all project users along with their rights (for example, access rights) for all CPUs in the project in the editor for users and roles of the project in the TIA Portal:

- Navigate to the "Security Settings > Users and roles" area in the project tree to manage users with their rights, for example, to control function rights.

The TIA Portal saves the assignment of the function rights of a CPU to user-defined roles and the assignment of these roles to users for each CPU. There are no system-defined roles with predefined function rights for CPUs. After loading the configuration, the user management becomes effective in the respective CPUs. After loading, every CPU "knows" who may access which service and execute specific functions.

The following settings can be made for a local project user:

- User name: Name of the local project user which must be used to log on to the project.
- Password: The password assigned by the administrator with which the project user can log on to the CPU. The project user can change the password later.
- Authentication method: For CPUs only possible via the password.
- Runtime timeout: Time span of inactivity after which a user is logged out from a device. You can use this timeout for your own web applications to have the application exit after a user-defined period of inactivity.
- Comment: Comment on the respective project user.
- Use the check box to the left of the user name to specify whether or not the project user should be downloaded to the CPU.

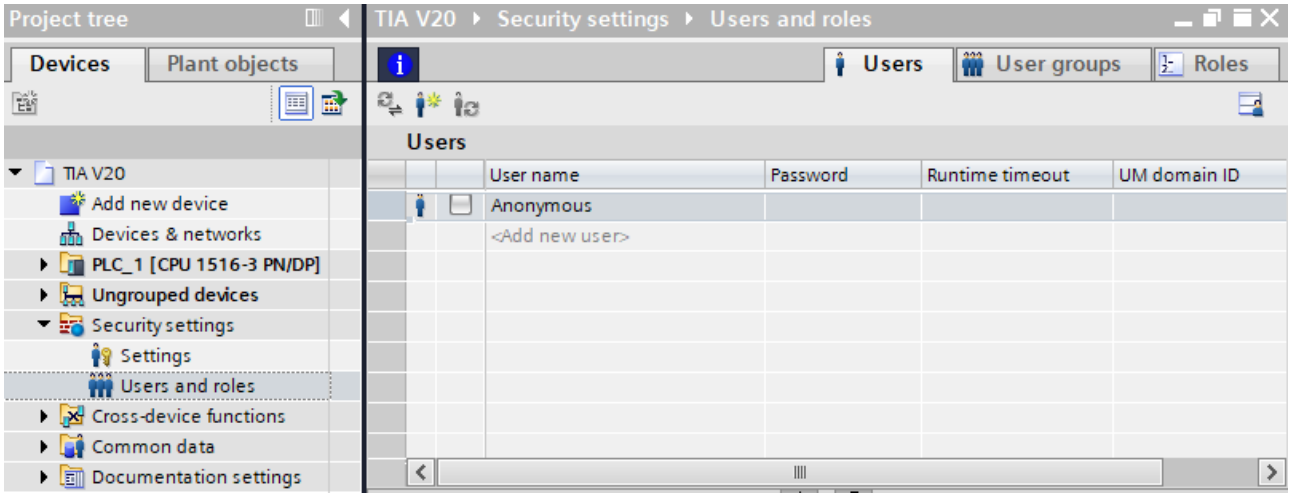


Figure 4-12 User management: Users and roles

You assign the roles and rights to the user.

Requirements:

CPU parameter assignment: To be able to set up users, roles and function rights for a CPU, the "Enable access control" option in the "Protection & Security > Access control" tab must be selected (which is the default). In addition, usage of the local user management must be activated in the "User management configuration".

Procedure:

In the following example, you assign the "admin" role to the user "User_1" with all rights for the web server:

1. Create a new local user in the "Users" tab, in the example "User_1", and assign a password.

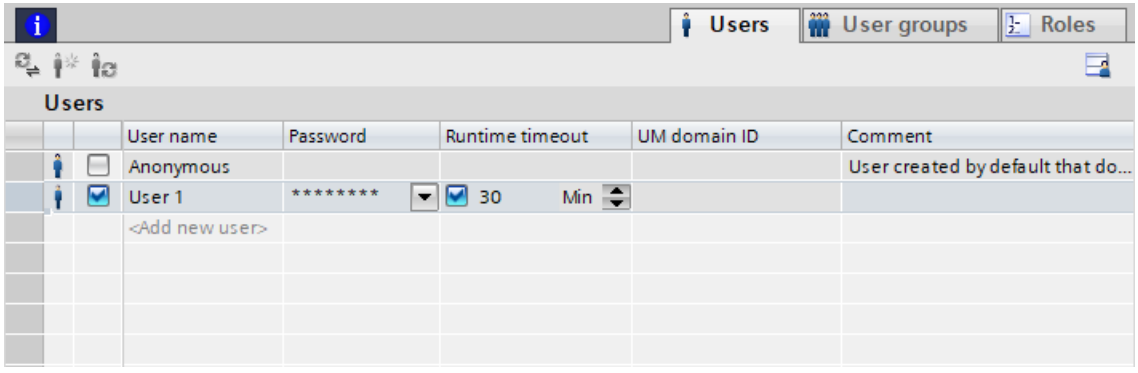


Figure 4-13 User management: Creating a local user

2. Define one or more roles in the "Roles" tab, the role "admin" in the example.

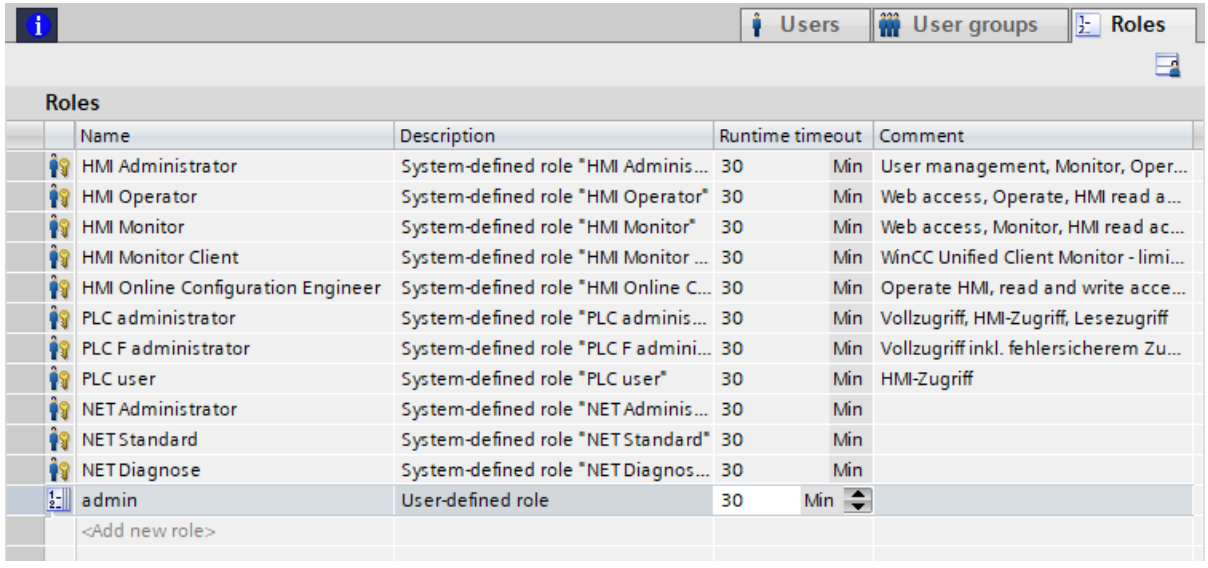


Figure 4-14 User management: Defining a role

3. Assign function rights for the Web server to the "admin" role. To do this, switch to the "Runtime rights" tab and select all function rights for the Web server for the example.

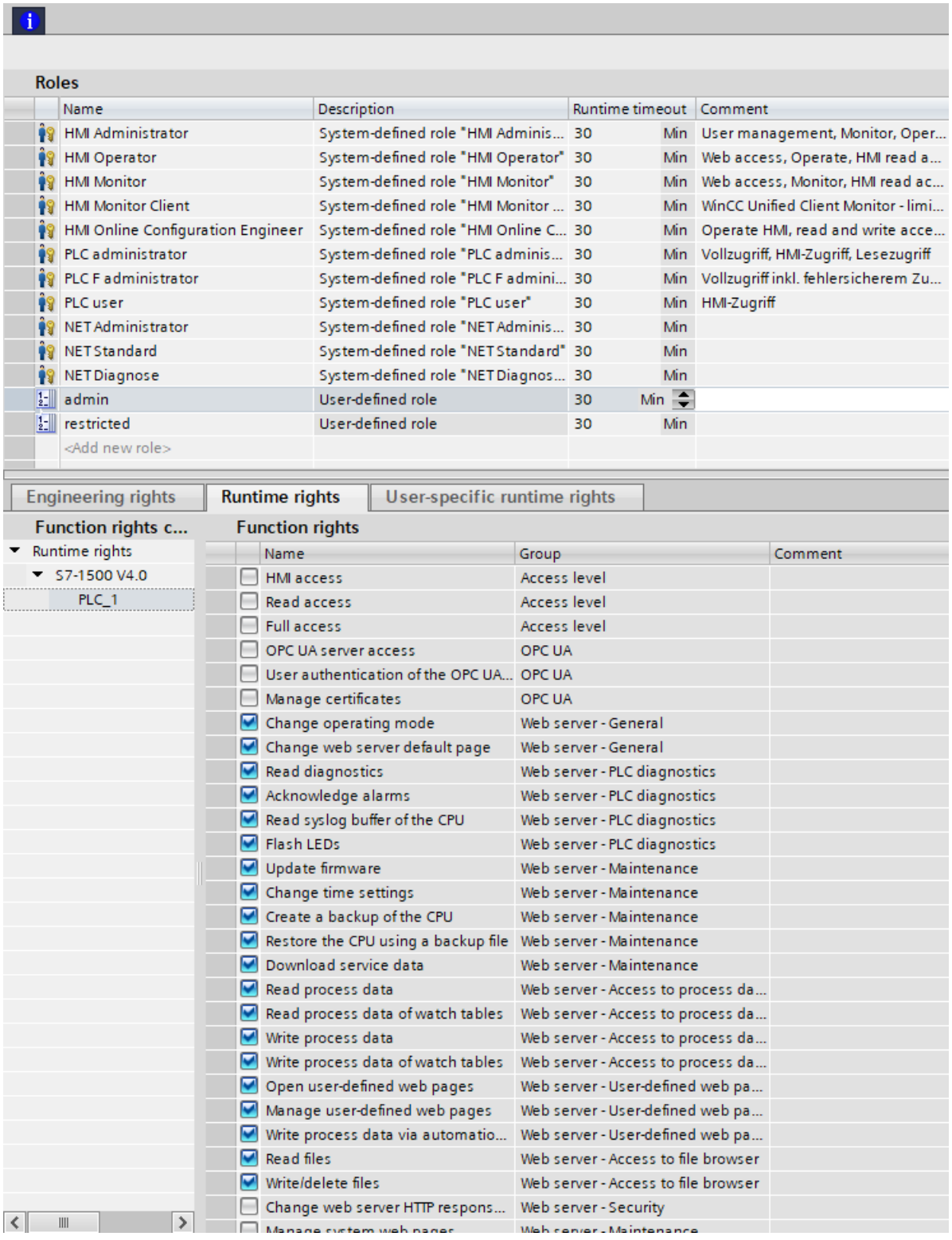


Figure 4-15 User management: Assigning rights to a role

4. At least one user must have full access to the CPU. Otherwise, the configuration cannot be compiled. Assign the "Full access" function right to the "admin" role.

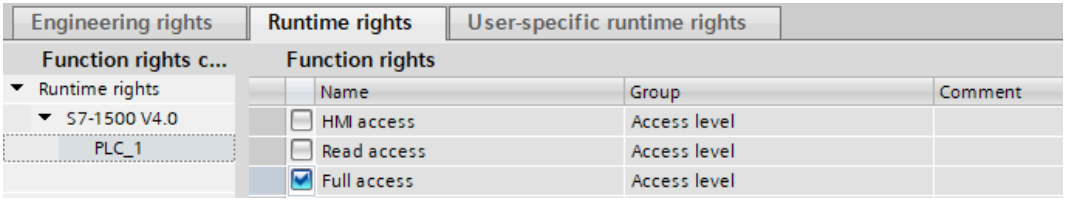


Figure 4-16 Setting up the full access access level

5. Assign one or more roles to the user "User_1" in the "Assigned roles" tab. In the example, you assign the "admin" role to the user "User_1". The roles and rights assigned to a user can be found in the corresponding tabs.

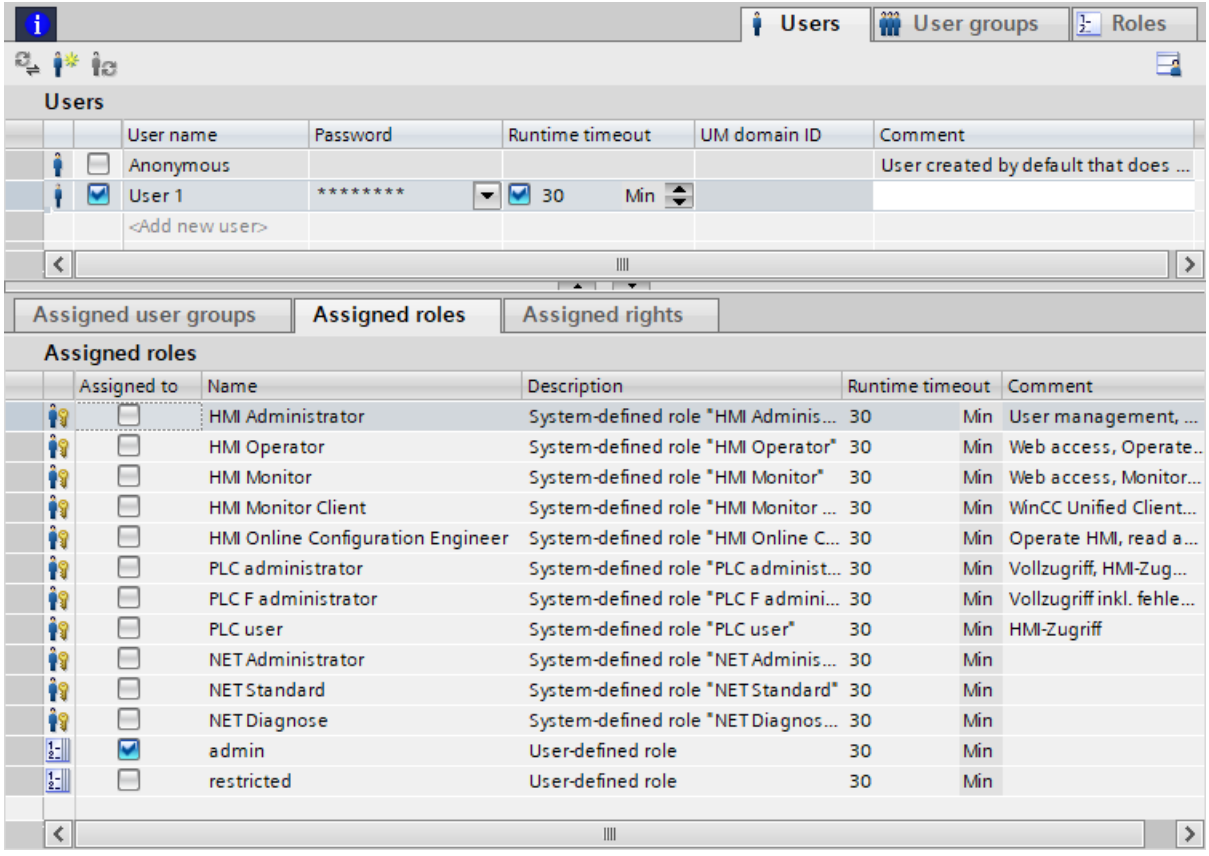


Figure 4-17 User management: Assigned role

6. Download the configuration to the CPU.

You can find more information on roles and rights in the online help for TIA Portal, keyword "Basics of user management in the TIA Portal".

If you delete a user in TIA Portal, their active user sessions are terminated.

Central user management

As of TIA Portal Version V20 and CPU firmware version V4.0/V40.0/V2.0, the CPUs support the central user management UMC (User Management Component).

The central user management allows system-wide, central management of users with an optional connection to Microsoft Active Directory.

With the software package UMC (User Management Component) you define and manage users and user groups on connected servers, across different software programs and devices. The authentication is implemented using UMC. UMC offers, among other things, a user management for CPU-specific function rights, such as for functions of the web server.

You set up linking between user/user group and assigned roles in the "Users and roles" editor in TIA Portal - as with the local user management.

During operation, users can be added to a group or removed from a group on the UMC server or their passwords can be changed without the CPU configurations having to be changed and loaded.

If you delete a user, their active user sessions are terminated.

Required information for the authorization via UMC

A CPU that has to authenticate the users for queries to the web server requires the following information to access the authentication service:

- The UMC server address

  Example: https://central.umc.testnet:443/ra

  (443 = Port for https, ra = Remote Authentication)
- A server ID (fingerprint of the server certificate).

Configure and apply central user management of the function rights for the web server of the CPU

You can find all the information on configuring and applying the central user management for CPUs in the section "Central user management" of the S7-1500, ET 200MP System Manual (https://support.industry.siemens.com/cs/ww/en/view/59191792).

Information on setting up (installing, configuring) a UMC server as well as on creating central users and groups is available in this application example (https://support.industry.siemens.com/cs/at/de/view/109780337/en).

Deactivating the user management

For CPUs with configured firmware version ≥ V3.1/V30.1/V1.0, you can disable access control in the hardware configuration in the CPU properties under "Protection & Security" > "Access control".

There is no user management in the project with deactivated access control. Users who are defined under "Users and roles" are not considered and authentication is not possible. The CPU itself creates an "Anonymous" user that has full access rights to the functions of the web server and the CPU.

If you deactivate local or central user management respectively, all the user sessions logged in locally or centrally are terminated.

NOTE

Take into account that deactivation of the access control is a security risk and can therefore not be recommended.

"Anonymous" user

As a non-logged-in user, you access web pages or the Web API as an "Anonymous" user. It does not matter in this case whether you have configured additional users. You activate the "Anonymous" user in the "Users" tab of the editor "Security settings > Users and roles".

The "Anonymous" user does not use any access authorizations by default and is specified without assignment of a password. However, you can assign all the function rights available in TIA Portal to it.

When the system web pages are called, these attempt by default to log into the system with the "Anonymous" user. The overview of the system web pages is opened automatically if one of the following conditions applies:

- An "Anonymous" user has been activated in TIA Portal.
- Function rights for the web server have been assigned to the "Anonymous" user.

The login page cannot be exited without successful authentication with a different user.

Number of users:

- Local user management: Minimum of 1 user and maximum of 100 users
- Central user management: Minimum of 1 user and maximum of 256 users or 50 user groups

NOTE

Because the "Anonymous" user is specified with assignment of a password in TIA Portal, pay close attention to the access rights you assign to this user.

The authorization on the web server as an "Anonymous" user is only foreseen for test purposes, commissioning, etc. In other words, when the system is not in productive operation. In this case, you have to ensure the security of the plant through other organizational measures, e.g. spatial protection.

Individual authorizations, such as the option to change the operating mode, may pose a safety risk.

Recommendation: When assigning security-relevant authorizations, create a user with password protection in TIA Portal.

⚠ WARNING

"Anonymous"user at an F-CPU

For an F-CPU, do **not** assign the access authorization "Perform changes as F-Admin" to the "Anonymous" user.

Make sure that you observe the warnings relating to this in the section "Restoring a backup of the safety program to an F-CPU" in the manual SIMATIC Safety - Configuring and Programming (https://support.automation.siemens.com/WW/view/en/54110126).

Selecting a user for a session

Always assign a user who has only the required authorizations for a session.

Example:

A user wants to display the new system web pages on an HMI panel. If the panel is used only for write-protected information, the user should not authenticate an administrator user who disposes of more function rights than are required.

A device that remains authenticated at the web server could be unsupervised so that an attacker could possibly carry out an unintended operation.

Rules for passwords

Passwords should always be more than 8 characters in length and contain uppercase and lowercase characters as well as special characters and numbers (?!+%$1234...). Computer keyboard character strings and words from the dictionary are unsuitable. Change the password regularly.

You can read the password policies of the CPU as of firmware version V3.1 with the Api.GetPasswordPolicy method. In the TIA Portal, you can read the policies and change these in the project tree via "Security settings" > "Settings" > "Password policies". You can find more information in the online help for the TIA Portal.

More information

More information on the user administration:

- with TIA Portal is available in the online help for the TIA Portal.
- for SIMATIC S7-1500 is available in the System Manual S7-1500, ET 200MP (https://support.industry.siemens.com/cs/ww/en/view/59191792), in the Protection section

System web pages

5

5.1 Getting started

Connecting to the Web server

Proceed as follows to access the web server:

1. Use STEP 7 to download a project in which the web server is activated to the CPU.
2. Optional: In the project tree in STEP 7, verify the "General settings" menu under the CPU in the "Web applications" folder and configure the system web pages optionally as default web pages of the web server.
3. Connect a display device (PG/PC, HMI, mobile terminal device) with the CPU or a communication module using a PROFINET interface.
   If you are working with WLAN, activate WLAN on the display device and establish a connection to the access point (e.g. SCALANCE W788-1RR or SCALANCE W784-1), which is in turn connected to the CPU.
4. Enter the IP address of the interface of the configured CPU which is connected to the client in the address bar of the web browser, for example:
   - https://192.168.3.141
   - If you have not configured the system web pages as default pages:
     https://192.168.3.141/~system

Result: The connection is established. The site to which you are directed depends on the access/function rights of the user.

You can find more information in the section "Setting up and calling a web server (Page 19)."

NOTE

Web pages for R/H-CPUs

The web pages are not available for the web server of the R/H-CPUs. Entering the IP address of the browser results in an error message.

The following section provides information about the different web pages using an S7-1500 CPU as an example.

Login page

In as far as the web server was activated and can be reached, you can access the new web pages of the web server.

The following figure shows the login page displayed by the web browser, if authentication is required.

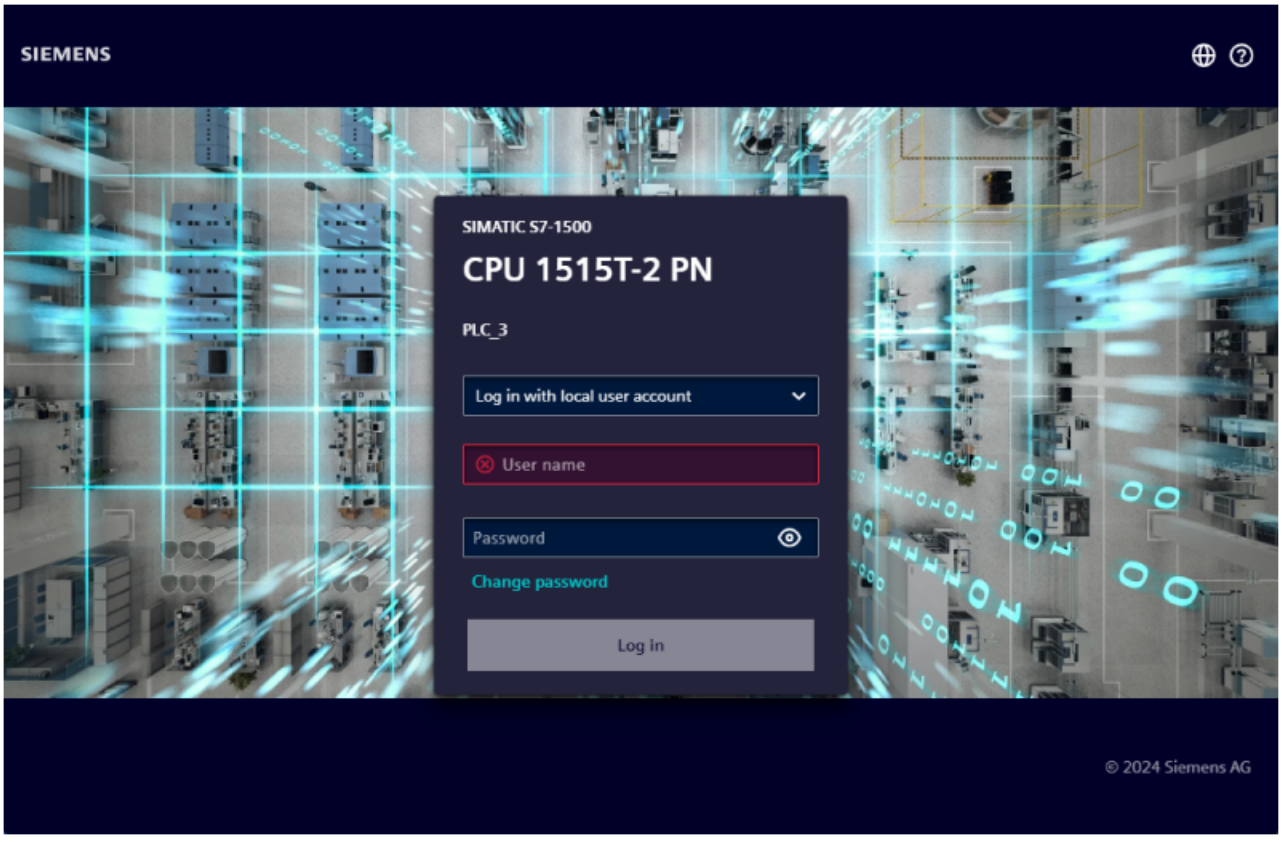


Figure 5-1 Login page

Follow the instructions for the Authentication (Page 53) to access the web server pages.

You can find more information in section System web pages and previous web pages (Page 26).

Overview of the system web pages

Table 5-1 System web pages of the CPUs as of firmware version V4.0/V40.0/V2.0

Web page	Submenus/Contents
Overview	Header with information on the currently logged-in users and language setting for the user interface General information, for example configured module name and module type
Diagnostics	• Contents of the diagnostic buffer of the CPU
User program	• Recipes • UserFiles (user files)
Alarming & Logging	• Alarms • Data logs
Maintenance	• File browser: Read out and process contents of the SIMATIC load memory
Information & legal	• Copyright note

More information

The information of the Web server Function Manual (https://support.industry.siemens.com/cs/ww/en/view/59193560), Edition 11/2023, continues to apply for the web pages that are **not** listed in the table.

Display

The following designs are available for the display of the web pages:

- Light: Dark print on a light background
- Dark: Light print on a dark background:

The design that is used depends on the system that the web pages display.

Time stamp

All time stamps are displayed by default based on the PLC local time.

5.1.1 Authentication

To use the full range of functions offered by the web server you require corresponding function rights. Log in with a user name and password specified in the user management in TIA Portal. You now have corresponding permissions to access the web pages released for this user.

You can find more information in the section "User management (Page 41)."

NOTE

After carrying out the actions planned by you, log out explicitly from the web server by clicking "Logout" in order to minimize the risk of unauthorized access.

NOTE

If your password has expired, you are informed when logging in that your password must be changed. As soon as you have changed your password, you can log into the web server again.

If your password expires soon, you are informed when logging in that your password must be changed soon.

Logging in to the web server

When the system web pages are called, the login page for the web server is opened on which you can log in. If the "Anonymous" user has at least one function right for the web server, the Start page (Page 58) of the web server is opened automatically.

NOTE

Logging in at the web server is required under the following conditions:

- No "Anonymous" user is activated in TIA Portal.
- No function rights for the web server have been assigned to the "Anonymous" user.

Proceed as follows to log into the web server:

1. Open the web browser on your display device.
2. Enter the IP address of the interface of the CPU which is connected to the client in the "Address" field of the web browser in the following format: https://a.b.c.d (example entry: https://192.168.3.141)
3. Enter your user name and your password in the login mask of the login page.

   In addition, you can choose between two login methods:

   – Logging in with a local user account
   – Logging in with a central user account

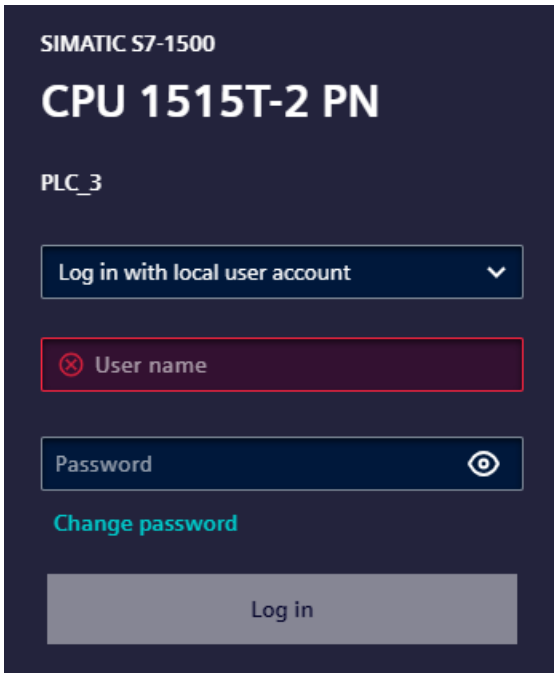


Figure 5-2 Logging in to the web server

4. Click "Login".

Result: You are in the web server overview The user currently logged in is displayed in the header.

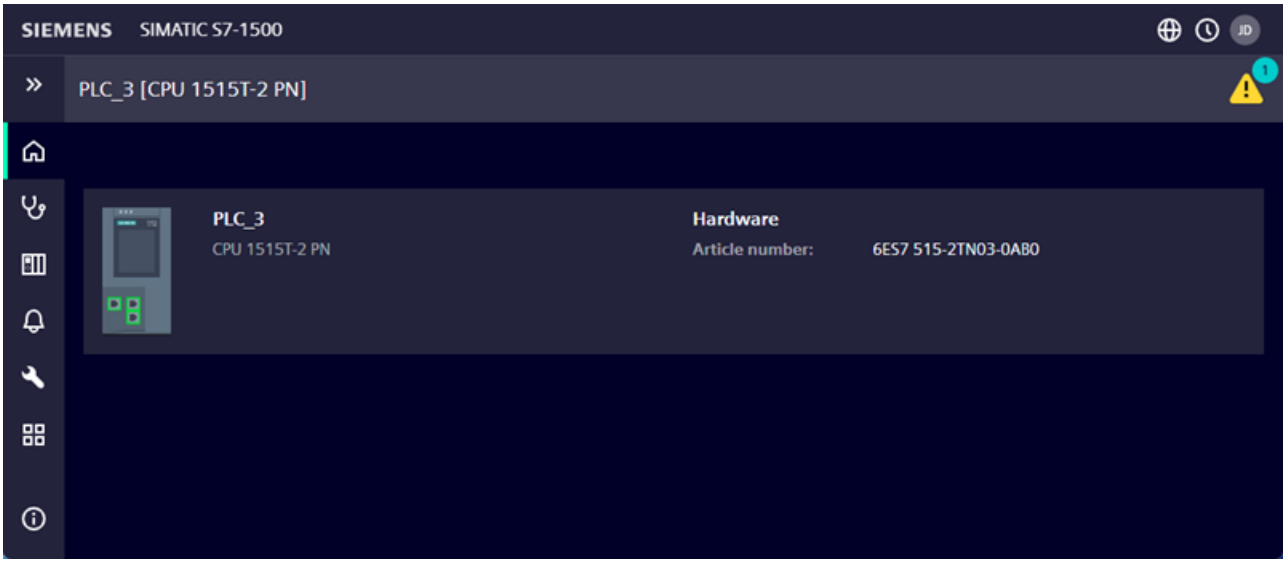


Figure 5-3 Overview of the web server

NOTE

The user is always shown the information for which they have authorizations. All further information is hidden.

Logging out from the web server

Proceed as follows to log out of the web server:

1. While you are logged in, click on your user name in the header of the web server.
2. Click "Logout".

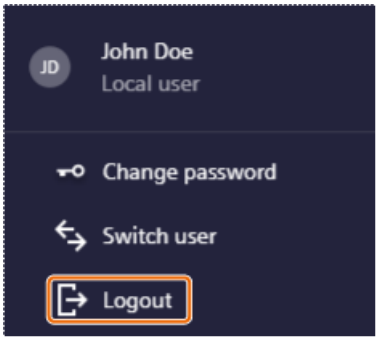


Figure 5-4 Logging out from the web server

Result: You have been logged out of the web server successfully and are on the login page.

Changing the user

Proceed as follows to change the user logged in at the web server:

1. While you are logged in, click on your user name in the header of the web server.
2. Click "Switch user".

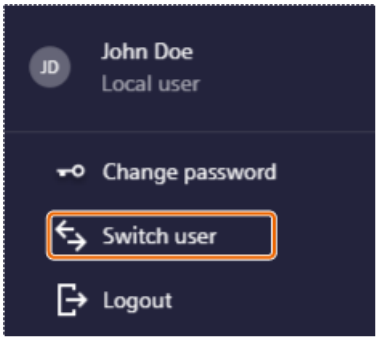


Figure 5-5 Changing the user

3. Enter a different user name and the corresponding password in the login mask in the popup window.
4. Click "Switch user".

Result: The logged-in user currently logged in is displayed in the header.

Change the password

The password can be changed before logging in on the login page as well as in the overview of the web server after logging in.

NOTE

The password can only be changed for local user accounts.

Changing the password on the login page

Proceed as follows to change the password on the login page:

1. Click "Change password".

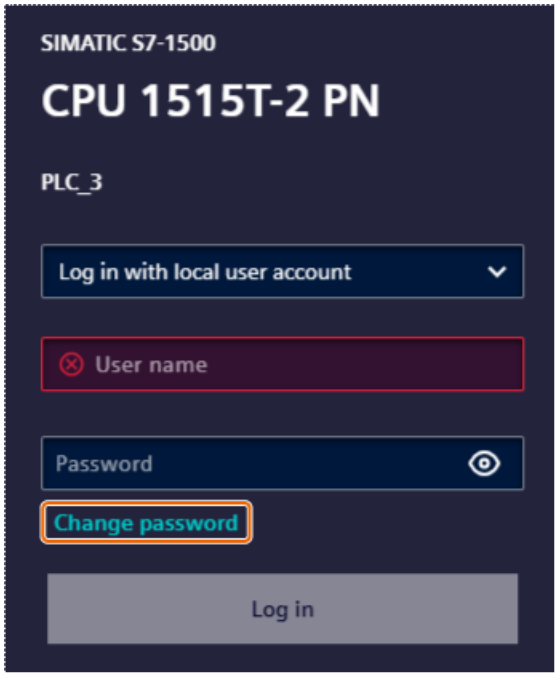


Figure 5-6 Changing the password on the login page

2. Enter your user name in the following window.
3. Click "Next".
4. In the following window enter your old password once as well as your new password twice in the corresponding input fields.
5. Click "Confirm".

NOTE

The password specifications, for example length and permitted characters, are displayed below the entry field and are confirmed with a green check mark when used correctly.

Result: The password has been changed and you are in the web server overview The user currently logged in is displayed in the header.

Changing the password in the logged-in state

Proceed as follows to change the password in the web server overview:

1. While you are logged in, click on your user name in the header of the web server.
2. Click "Change password".

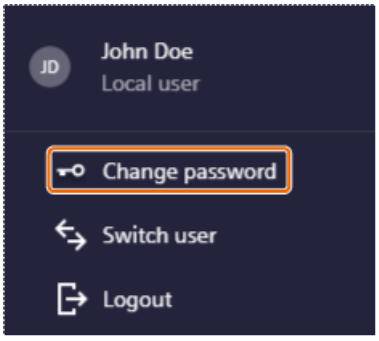


Figure 5-7 Changing the password in the overview

3. In the popup window, enter your old password once as well as your new password twice in the corresponding input fields.
4. Click "Confirm".

NOTE

The password specifications, for example length and permitted characters, are displayed below the entry field and are confirmed with a green check mark when used correctly.

The password specifications are configurable under "Users and roles" in TIA Portal. You can find more information in section User management (Page 41).

Result: The password has been changed and you are in the web server overview The user currently logged in is displayed in the header.

5.1.2 Operating state and messages

Overview

The secondary header of the homepage contains the following information:

- ① Configured module name and type
- ② Number of messages

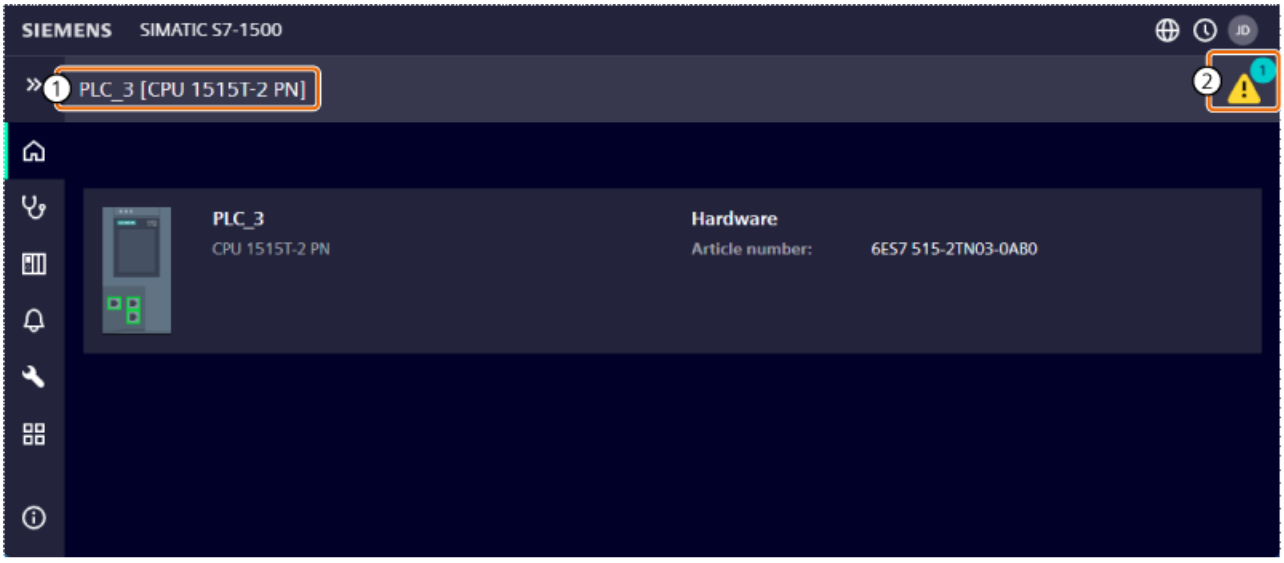


Figure 5-8 Secondary header overview

See also

Alarms (Page 65)

5.2 Overview

Overview

After you have logged in (see section "Authentication (Page 53)"), you are in the web server overview.

The following contents and information is provided in the overview:

- ① Header with information on the currently logged-in users as well as time zone and language settings
- ② Secondary header with information on the selected CPU type, see section "Operating state and messages (Page 58)"
- ③ Information on the hardware used

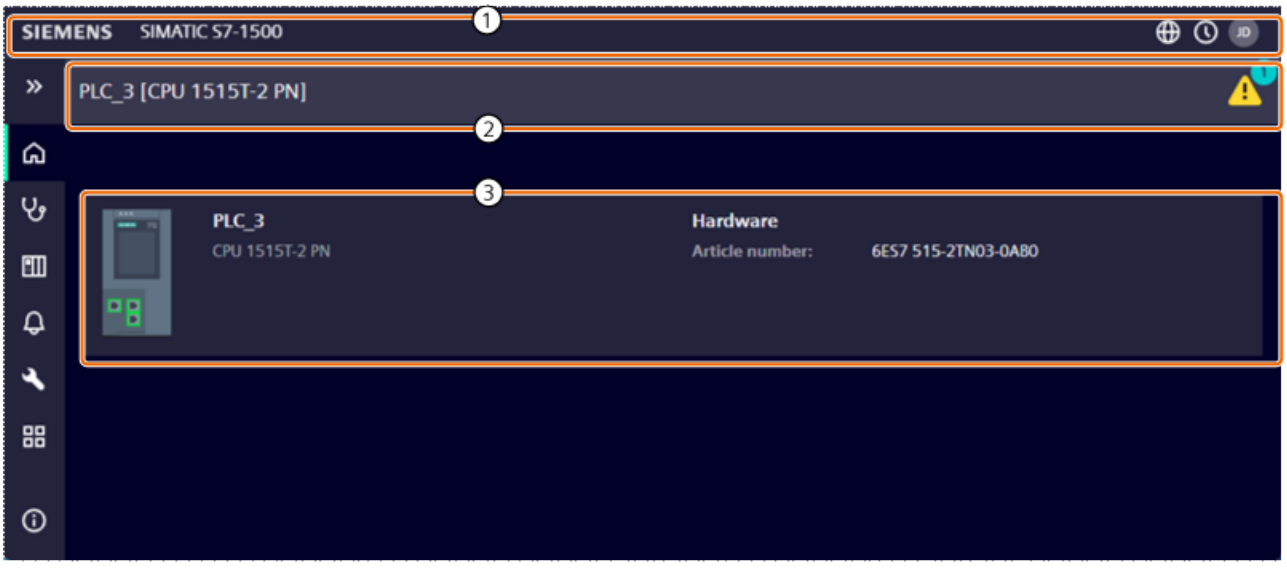


Figure 5-9 Overview

Navigation

With the "Menu" button, you open the navigation area of the web server. Here you can access the following web pages and information:

- Diagnostics (Page 59)
- User program (Page 62)
- Alarming & Logging (Page 64)
- Maintenance (Page 67)

The contents of the web pages are described in the following sections.

5.3 Diagnostics

Overview

The following web pages and information are available under the "Diagnostics" menu item:

- Diagnostics buffer (Page 60)

5.3.1 Diagnostics buffer

Overview

With "Diagnostics" > "Diagnostics buffer", you open the diagnostics buffer of the CPU and load the diagnostics buffer into the web server.

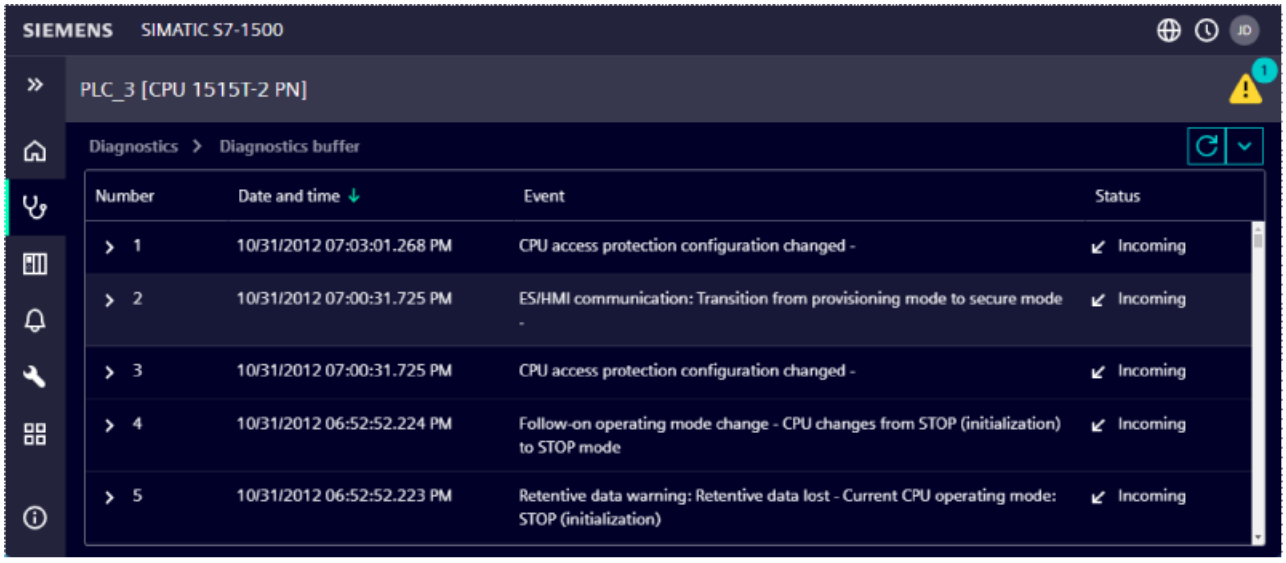


Figure 5-10 "Diagnostics buffer" overview

NOTE

Load strategy for diagnostic buffer entries

Adjust the load strategy to the available performance. At best, all diagnostic buffer entries are displayed.

It can be possible that not all diagnostic buffer entries are displayed. To load further diagnostic buffer entries, scroll to the end of the displayed diagnostic buffer entries and select one of the following options:

- Reload last {{count}} entries
- Reload complete buffer

Note that the diagnostic events are displayed in the project language of the STEP 7 project that is assigned to the current web server interface language. You can find out how to assign project languages to interface languages in the section "Language settings (Page 39)".

Opening details of a diagnostic buffer entry

Click a diagnostic buffer entry to view its details.

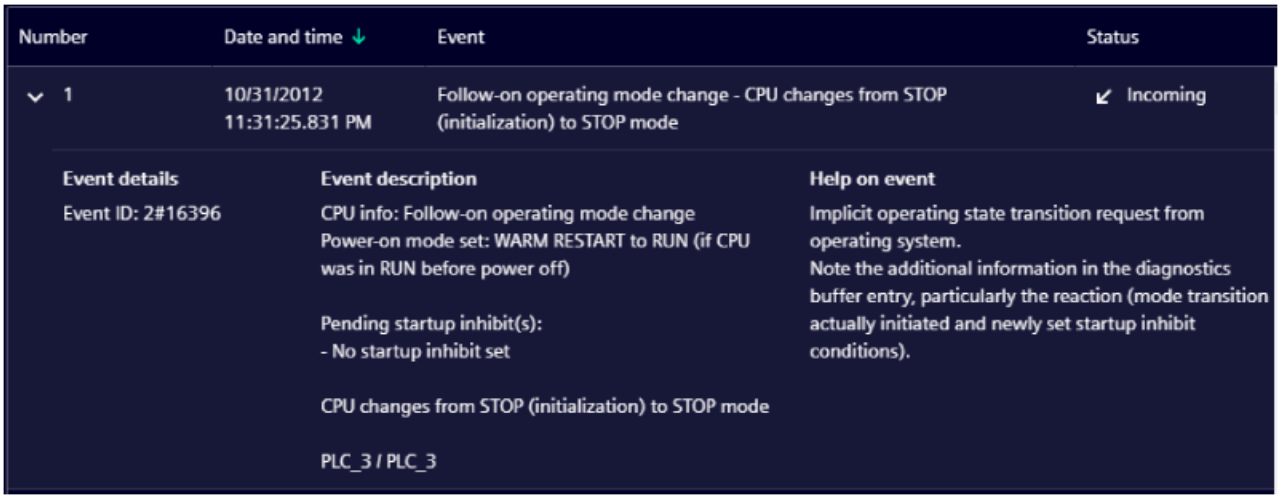


Figure 5-11 Details of a diagnostics buffer entry

Updating the diagnostics buffer

With the "Refresh" button, you update the diagnostics buffer and load new incoming diagnostic buffer entries.

Notifications

If a new diagnostics buffer entry comes in, an asterisk (*) is displayed above the "Refresh" button. Position the cursor above the button to see when the new diagnostic buffer entry came in.

NOTE

The asterisk (*) above the "Refresh" button is displayed with a delay after a new diagnostics buffer entry has come in.

NOTE

The diagnostic buffer is not updated automatically when new diagnostic buffer entries come in

5.4 User program

Overview

The following web pages and information are available under the "User program" menu item:

- Recipes
- User files

Creating a folder for recipes

Proceed as follows to create a folder for recipes:

1. Navigate to "User program" > "Recipes".

   If no folder has been created for recipes yet, the view displays the "Create folder" message.

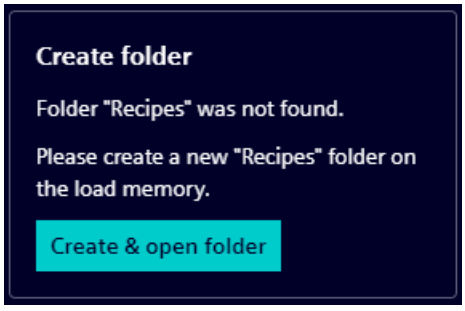


Figure 5-12 "Create folder" message

   If you do not have the required authorizations to create a folder in the SIMATIC load memory, the view shows "Folder "Recipes" was not found".

Figure 5-13 "Folder "Recipes" was not found" message

   Set the notifications in the TIA project or contact your administrator.

2. Click the "Create & open folder" button.

Result: You are forwarded to "Maintenance" > "File browser" > "Load memory" > "Recipes" and the created folder is opened.

Creating a folder for user files

Proceed as follows to create a folder for user files:

1. Navigate to "User program" > "User files".

   If no folder has been created for user files yet, the view displays the "Create folder" message.

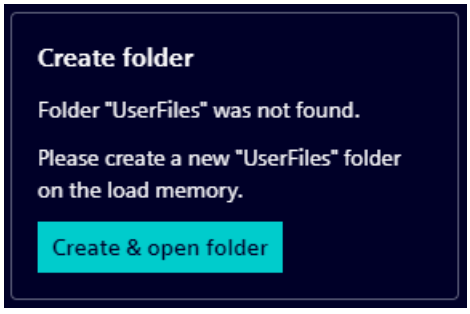


   Figure 5-14 "Create folder" message

   If you do not have the required authorizations to create a folder in the SIMATIC load memory, the view shows "Folder "UserFiles" was not found".

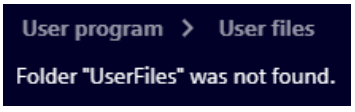


   Figure 5-15 "Folder "UserFiles" was not found" message

   Set the notifications in the TIA project or contact your administrator.

2. Click the "Create & open folder" button.

Result: You are forwarded to "Maintenance" > "File browser" > "Load memory" > "UserFiles" and the created folder is opened.

5.5 Alarming & Logging

Overview

The following web pages and information are available under the "Alarming & Logging" menu item:

- Alarms (Page 65)
- Data logs (Page 66)

Creating a folder for data logs

Proceed as follows to create a folder for data logs:

1. Navigate to "Alarms" > "Data logs".

   If no folder has been created for data logs, the view displays the message "Create folder".

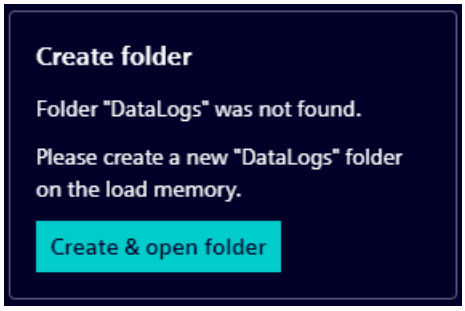


Figure 5-16 "Create folder" message

   If you do not have the required authorizations to create a folder in the SIMATIC load memory, the view shows "Folder "Data logs" was not found".

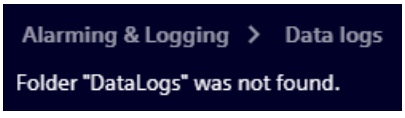


Figure 5-17 "Folder "Data logs" was not found" message

   Set the authorizations in the TIA project or contact your administrator.

2. Click the "Create & open folder" button.

Result: You are forwarded to "Maintenance" > "File browser" > "Load memory" > "DataLogs" and the created folder is opened.

Downloading and deleting files in the folder for data logs

You have the following options for downloading files in the folder for data logs:

Downloading and deleting a single file

1. In the row of the desired file, click the "Download and delete" button.

Downloading and deleting several files

1. In the current folder, click next to the desired files on the associated check boxes to select these. The number of selected files is displayed above the "Expand" button.
2. Click the "Download" button. If this button is not displayed in the top line, click on the "Expand" button and select the "Download" function in the drop-down menu.
3. Click the "Delete" button. If this button is not displayed in the top line, click on the "Expand" button and select the "Delete" function in the drop-down menu.

Result: The files are downloaded one after the other and deleted in the second step. A popup window is displayed with feedback about successful or failed downloading of the files.

5.5.1 Alarms

Overview

Read out the message buffer to obtain compact information for fault analysis. This is the most effective method to get an overview of the pending faults.

The "Alarms" web page contains the alarm buffer with time specification, alarm text and status as well as the following possibilities:

- ① Viewing of details of a message via drop-down function
- ② Acknowledging of a message

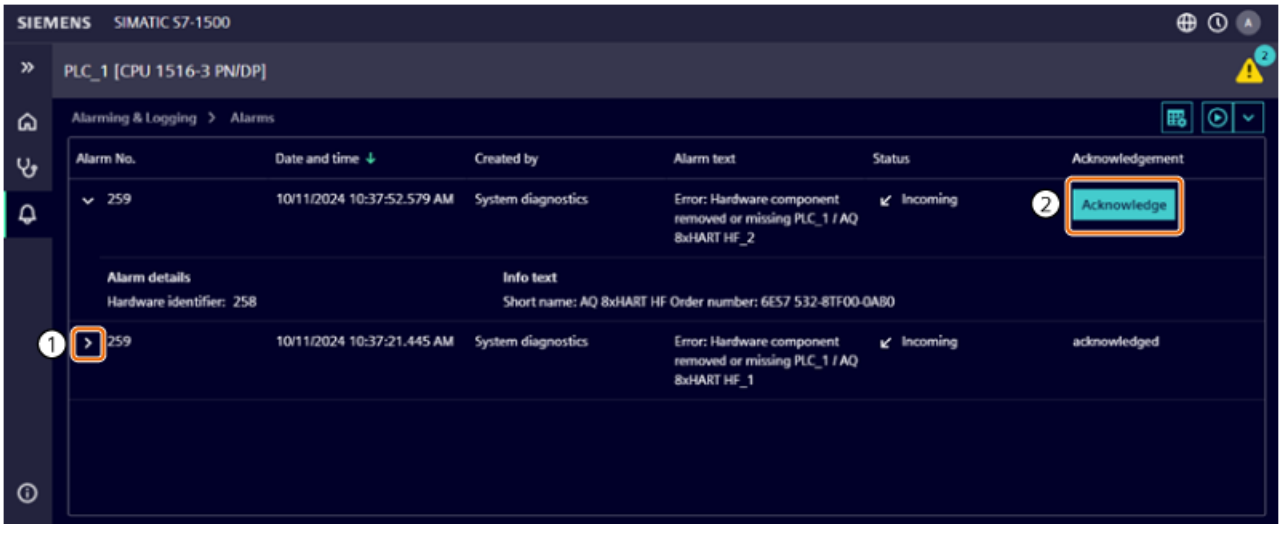


Figure 5-18 "Alarms" overview

Messages are displayed by default in descending chronological order with date and time.

The "Alarm text" column contains the entry of configured alarm texts of the respective error definitions.

The alarm texts are displayed in the project language of the STEP 7 project that is assigned to the current interface language of the web server. You can find out how to assign project languages to interface languages in the section "Language settings (Page 39)".

Viewing and acknowledging of messages

Proceed as follows to view and acknowledge messages:

1. Open the "Alarms" view.
2. Click the "Acknowledge" button of all desired messages to acknowledge these.

   A detailed view is optionally available via the "Expand" button.

Freezing of the alarm buffer

Proceed as follows to freeze the message buffer:

1. Open the "Alarms" view.
2. Click the "Pause" button.

Result: As long as the message buffer is frozen, the list of messages is not updated. Only the messages already available are displayed.

If new alarms are available at a frozen alarm buffer, the number of new alarms is displayed above the "Play" button. When you click the button, all the active and new messages are displayed in the list.

5.5.2 Data logs

Overview

All the created data logs can be viewed on the "Data logs" web page.

The data logs can be sorted according to individual parameters in ascending or descending order.

For this purpose, click on one of the parameters in the column header:

- Name
- Last modified
- Size

You download the relevant data log file by clicking the file name.

The "Active" column shows whether the respective data log file is used (is active) or not in the user program.

When the data log file is active, you can call (download) and empty the relevant data log file by clicking the icon. The file must be closed to this purpose. The empty data log file is still included in the list of data logs.

You delete the data log file by clicking the icon in the "Delete" column. The file must be closed to this purpose.

You close an opened data log file in STEP 7 by using the "DataLogClose" instruction.

5.6 Maintenance

Overview

You can find the web page File browser (Page 67) under the "Maintenance" menu item.

5.6.1 File browser

Introduction

The web browser displays the content of the SIMATIC load memory on the web page "Maintenance" > "File browser" > "Load memory". This means, for example, that you can read and edit the log files generated by the CPU without having to use STEP 7.

The file browser lists all the existing files and folders that are located in the SIMATIC load memory.

You can download, delete, rename and upload the files as well as create, delete and rename the directories.

NOTE

There are restrictions for accessing the Recipes, Data logs, and User files folders. For example, subfolders cannot be created.

NOTE

Exception system files

The system files include the job file and all special folders including their contents to which the job file refers. System files are not displayed, and cannot be changed or deleted.

Overview

On the "Maintenance" > "File browser" > "Load memory" web page, you can edit and manage files and folders and find the following information:

- ① Path to the current folder
- ② Files in the current folder
- ③ Structure of folders

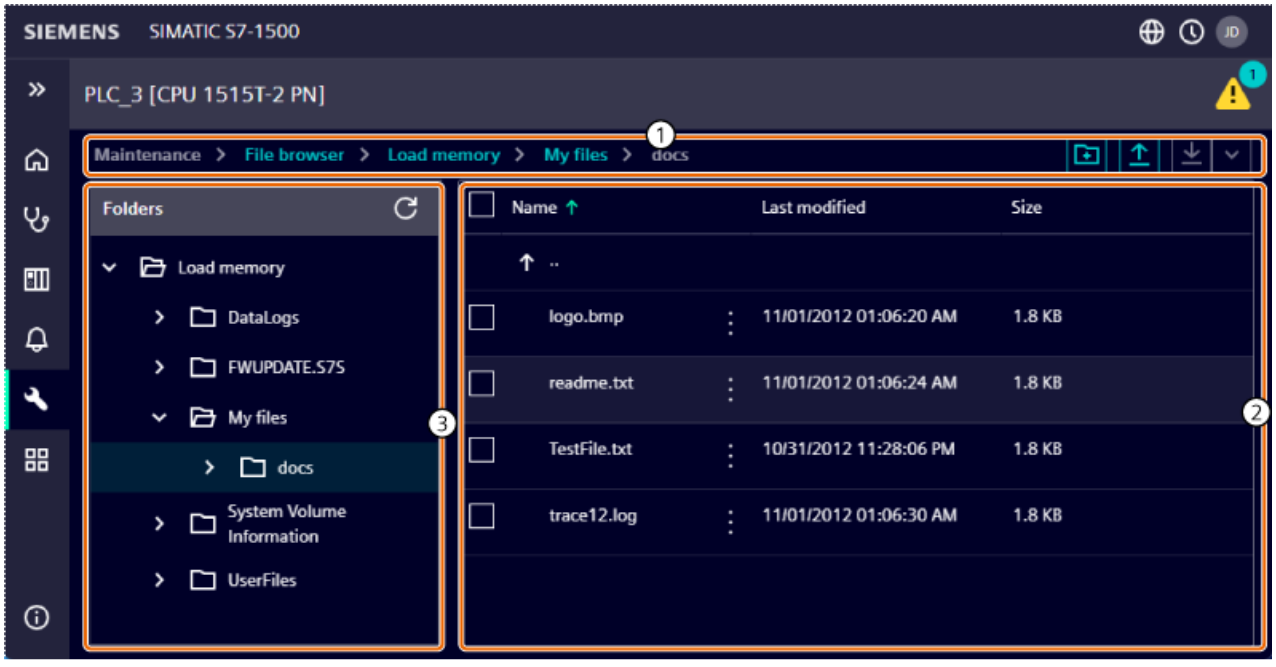


Figure 5-19 "File browser" overview

Uploading a file

Proceed as follows to upload a file:

1. In the file structure select the folder into which you want to upload the file.
2. Click the "Upload" button.
3. A browser-specific dialog box opens. Select the desired file.

Result: After uploading the current folder shows the uploaded file. A popup window is displayed with feedback about successful or failed uploading of the file.

Moving a file

Proceed as follows to move a file:

1. In the current folder, click the button next to the desired file.
2. A context menu opens. Click "Move to".
3. A dialog box opens. Navigate to the desired folder and confirm with "Move".

Result: The current folder no longer shows the moved file. A popup window is displayed with feedback about successful or failed moving of the file.

Moving several files

Proceed as follows to move several files:

1. In the current folder, click next to the desired files on the associated check boxes. The number of selected files is displayed above the "Expand" button.
2. Click on the "Expand" button and click on "Move to".
3. A dialog box opens. Navigate to the desired folder and confirm with "Move".

Result: The current folder no longer shows the moved files. A popup window is displayed with feedback about successful or failed moving of the file.

Renaming a file

Proceed as follows to rename a file:

1. In the current folder, click the button next to the desired file.
2. A context menu opens. Click "Rename".
3. A dialog box opens. Enter the required file name and confirm with "Rename".

NOTE

Invalid character

If you use an invalid character, you are informed of this by means of a text box under the "File name" text field, and you cannot confirm with "Rename". Delete invalid characters in order to continue.

Result: The current folder shows the new file name. A popup window is displayed with feedback about successful or failed renaming of the files.

Creating a folder

Proceed as follows to create a folder:

1. In the file structure select the folder in which you want to create the new folder.
2. Click the button.
3. A dialog box opens. Enter the required folder name and confirm with "Create".

Result: The current folder shows the new folder. A popup window is displayed with feedback about successful or failed creation of the folder.

Deleting a file or folder

Proceed as follows to delete a file or folder:

1. In the current folder, click the button next to the desired file or the desired folder.
2. A context menu opens. Click "Delete".

NOTE

Only empty folders can be deleted. If you want to delete a folder with contents, first delete all the files and folders in it.

3. A dialog box opens. Click the "Delete" button.

Result: The current folder no longer shows the file or folder. A popup window is displayed with feedback about successful or failed deletion.

Deleting multiple files or folders

To delete multiple files or folders, follow these steps:

1. In the current folder, click next to the desired files or folders on the associated check boxes. The number of selected files is displayed above the "Expand" button.
2. Click the "Delete" button. If this button is not displayed in the top line, click on the "Expand" button and select the "Delete" function in the drop-down menu.

NOTE

Only empty folders can be deleted. If you want to delete a folder with contents, first delete all the files and folders in it.

3. A dialog box opens. Navigate to the desired folder and confirm with "Delete".

Result: The current folder no longer shows the moved files and folders. A popup window is displayed with feedback about successful or failed deletion of files and folders.

API (Application Programming Interface) 6

Web API

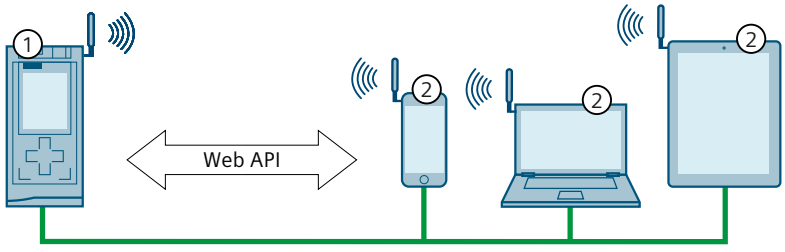
The CPU offers you a web-based API (Web API) as an interface for reading and writing CPU data.

The Web API enables you to:

- Communicate with the Web server of the CPU via script and programming languages
- Implement web applications at the latest state-of-the-art technology
- Create web applications that connect to multiple CPUs at the same time, for example, to create dashboards that visualize the status of multiple CPUs
- Loading web applications on to the web server of the CPU and making them available to the user. See section Web applications that can be loaded by the user (Page 117).

Relation between CPU, Web API and end devices

The following image shows an example of the Web API between CPU and end device.



① CPU

② Terminal devices

Figure 6-1 Web API

Communication between the CPU and the terminal device takes place via PROFINET or WLAN integration.

NOTE

Security information

Note that the graphic only shows the role of the Web API between CPU and end device.

For the correct setup of a secure WLAN connection, observe the security information in the section Safety instructions (Page 16).

6.1 Supported clients

Requirements

The Web API can only be used for CPUs as of firmware version V2.8 for the following systems:

- Standard CPUs, F-CPUs and T-CPUs of the S7-1500 automation system
- CPUs of the ET 200pro distributed I/O system
- Standard CPUs, F-CPUs and T-CPUs of the distributed I/O system ET 200SP
- Standard CPUs, F-CPUs and T-CPUs of the S7-1500 Software Controller
- As of firmware version V3.1, the R/H-CPUs of the S7-1500 automation system
- As of firmware version V1.0 the S7-1500 Virtual Controller

The following requirements must be fulfilled before you can use the Web API:

- You have assigned the correct firmware version (≥ V2.8) in the Hardware catalog of STEP 7.
- You have created and configured a project and downloaded it to the CPU.
- You have ensured that the following check box is selected in STEP 7:
  - Activate Web server on this module

Overview of the Web API methods depending on the firmware version of the CPU

The Web API is being continuously extended. The following table shows which mechanisms and methods you can use on which CPU as of which firmware version:

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
Basic functions				
Api.Login (Page 85)	V2.8	V2.8	V3.1	V1.0
Api.Logout (Page 89)	V2.8	V2.8	V3.1	V1.0
Api.GetPermissions (Page 89)	V2.8	V2.8	V3.1	V1.0
Api.ChangePassword (Page 91)	V3.1	V3.1	V3.1	V1.0
Api.GetPasswordPolicy (Page 92)	V3.1	V3.1	V3.1	V1.0
Api.GetAuthenticationMode (Page 94)	V3.1	V3.1	V3.1	V1.0
Api.GetSessionInfo (Page 95)	V4.0	V4.0	V4.0	V2.0
Api.Browse (Page 98)	V2.8	V2.8	V3.1	V1.0
Api.Version (Page 99)	V2.8	V2.8	V3.1	V1.0
Api.Ping (Page 99)	V2.8	V2.8	-	V1.0
Api.GetCertificateUrl (Page 100)	V2.8	V2.8	-	V1.0
Api.GetQuantityStructures (Page 100)	V3.1	V3.1	V3.1	V1.0

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
Ticket mechanism				
Api.BrowseTickets (Page 106)	V2.9	V2.9	V3.1	V1.0
Api.CloseTicket (Page 108)	V2.9	V2.9	V3.1	V1.0

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
Reading and writing user-configurable HTTP response headers				
WebServer.ReadResponseHeaders (Page 112)	V4.0	V4.0	-	V2.0
WebServer.ChangeResponseHeaders (Page 114)	V4.0	V4.0	-	V2.0

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
Setting the Web server default page				
WebServer.SetDefaultPage (Page 115)	V3.1	V3.1	-	V1.0
WebServer.ReadDefaultPage (Page 117)	V3.1	V3.1	-	V1.0

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
Web applications that can be loaded by the user				
WebApp.Create (Page 125)	V2.9	V2.9	-	V1.0
WebApp.Delete (Page 126)	V2.9	V2.9	-	V1.0
WebApp.Rename (Page 127)	V2.9	V2.9	-	V1.0
WebApp.Browse (Page 128)	V2.9	V2.9	-	V1.0
WebApp.SetState (Page 131)	V2.9	V2.9	-	V1.0
WebApp.SetDefaultPage (Page 132)	V2.9	V2.9	-	V1.0
WebApp.SetNotFoundPage (Page 133)	V2.9	V2.9	-	V1.0

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
WebApp.SetNotAuthorizedPage (Page 135)	V2.9	V2.9	-	V1.0
WebApp.BrowseResources (Page 136)	V2.9	V2.9	-	V1.0
WebApp.CreateResource (Page 138)	V2.9	V2.9	-	V1.0
WebApp.DeleteResource (Page 140)	V2.9	V2.9	-	V1.0
WebApp.RenameResource (Page 141)	V2.9	V2.9	-	V1.0
WebApp.DownloadResource (Page 142)	V2.9	V2.9	-	V1.0
WebApp.SetResourceVisibility (Page 144)	V2.9	V2.9	-	V1.0
WebApp.SetResourceETag (Page 145)	V2.9	V2.9	-	V1.0
WebApp.SetResourceMediaType (Page 147)	V2.9	V2.9	-	V1.0
WebApp.SetResourceModificationTime (Page 148)	V2.9	V2.9	-	V1.0
WebApp.SetVersion (Page 149)	V4.0	V4.0	-	V2.0
WebApp.SetUrlRedirectMode (Page 151)	V4.0	V4.0	-	V2.0

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
Reading and writing process data				
PlcProgram.Read (Page 157)	V2.8	V2.8	-	V1.0
PlcProgram.Write (Page 159)	V2.8	V2.8	-	V1.0
PlcProgram.DownloadProfilingData (Page 162)	V3.1	V3.1	V4.0	V1.0
PlcProgram.Browse (Page 167)	V2.8	V2.8	-	V1.0

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
Reading and changing the operating mode				
Plc.ReadOperatingMode (Page 174)	V2.9	V2.9	V4.0	V1.0
Plc.RequestChangeOperatingMode (Page 176)	V2.9	V2.9	V4.0	V1.0
Plc.ReadModeSelectorState (Page 177)	V3.1	V3.1	V4.0	V1.0

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
Entering time settings via Web API				
Plc.ReadSystemTime (Page 178)	V3.0	V3.0	-	V1.0
Plc.SetSystemTime (Page 179)	V3.1	V3.1	-	V1.0
Plc.ReadTimeSettings (Page 180)	V3.0	V3.0	-	V1.0
Plc.SetTimeSettings (Page 183)	V3.1	V3.1	-	V1.0

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
Reading diagnostics and service data				
Project.ReadLanguages (Page 186)	V3.1	V3.1	-	V1.0
Alarms.Browse (Page 189)	V3.1	V3.1	-	V1.0
Alarms.Acknowledge (Page 196)	V3.1	V3.1	-	V1.0
DiagnosticBuffer.Browse (Page 197)	V3.1	V3.1	-	V1.0
Modules.DownloadServiceData (Page 202)	V3.1	V3.1	-	V1.0

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
Technology/Motion Control				
Technology.BrowseObjects (Page 204)	V4.0	V4.0	-	V2.0
Technology.Read (Page 206)	V4.0	V4.0	-	V2.0

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
Backing up and restoring the configuration				
Plc.CreateBackup (Page 207)	V3.0	V3.0	-	-
Plc.RestoreBackup (Page 209)	V3.0	V3.0	-	-

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
Accessing contents of the SIMATIC load memory				
Files.Browse (Page 213)	V3.0	V3.0	V3.1	V1.0
Files.Download (Page 216)	V3.0	V3.0	V3.1	V1.0
Files.Create (Page 218)	V3.0	V3.0	V3.1	V1.0
Files.Rename (Page 219)	V3.0	V3.0	V3.1	V1.0
Files.Delete (Page 221)	V3.0	V3.0	V3.1	V1.0
Files.CreateDirectory (Page 222)	V3.0	V3.0	V3.1	V1.0
Files.DeleteDirectory (Page 224)	V3.0	V3.0	-	V1.0
DataLogs.DownloadAndClear (Page 225)	V3.0	V3.0	V3.1	V1.0

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V F as of firmware version
Reading information from SIMATIC Safety				
Failsafe.ReadRuntimeGroups (Page 226)	-	V3.1	V4.0	V1.0
Failsafe.ReadParameters (Page 228)	-	V3.1	V4.0	V1.0

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
Reading CPU information				
Plc.ReadCpuType (Page 230)	V4.0	V4.0	V4.0	V2.0
Plc.ReadStationName (Page 231)	V4.0	V4.0	V4.0	V2.0
Plc.ReadModuleName (Page 231)	V4.0	V4.0	V4.0	V2.0

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
Reading information from a redundant system				
Redundancy.ReadSystemInformation (Page 232)	-	-	V4.0	-

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version	S7-1500V as of firmware version
Redundancy.ReadSystemState (Page 235)	-	-	V4.0	-
Redundancy.RequestChangeSystemState (Page 236)	-	-	V4.0	-
Redundancy.ReadSyncupProgress (Page 237)	-	-	V4.0	-

API endpoint

As an RPC protocol, JSON-RPC V2.0 is based on HTTP. The Web API can be reached via POST requests to the following URL:

```
https://[ip]/api/jsonrpc
```

The Web API supports bulk operations as defined in JSON-RPC 2.0. While bulk operations are not explicitly limited by a fixed number of method calls, there is a limit for the HTTP request body. The limit differs depending on the firmware version of the CPU:

- Limit of 64 KB for CPUs with firmware version ≤ V3.0
- Limit of 128 KB for CPUs as of firmware version ≥ V3.1

As of firmware version V3.1 you can use the API method Api.GetQuantityStructures to read out the limit.

An example of the required structure of an HTTP request and HTTP response for successfully making an API request can be found in the section Web API integration (Page 79).

Compatibility with regard to future extensions of the Web API

The order of attributes within a JSON object does not affect API clients.

Web server responses to requests via the Web API may be extended with new JSON attributes in future firmware versions, e.g. to enrich results with more details.

Error codes of possible API error messages based on JSON-RPC may change in future firmware versions, if applicable, and existing error messages may be made more precise.

NOTE

To check if your API request was successful, first check if the request was successful in general. You can evaluate the JSON-RPC error codes for detailed information.

Textual error information only provides information. If you want to implement error evaluation in a way specific to an application, use the corresponding numeric error codes of the error message.

The precision of floating point values may differ in the display from other clients, such as the TIA Portal.

Bulk requests are always processed in descending order, whereby the individual requests are contained in the HTTP request body in descending order.

If an API method includes time stamps, these time stamps are always returned based on UTC time. Examples of this are the time stamps of the Files.Browse and Api.BrowseTickets

methods. With the API methods Plc.ReadSystemTime and Plc.ReadTimeSettings, you can read out the system time and determine the CPU local time.

Supported clients

The products and versions listed in the following table were tested for the Web API. The column "As of Version" lists the tested version as of which the clients are supported:

Product *	As of Version	Supported functions
Chrome-based Desktop web browser (e.g. Google Chrome) (https://chromium.woolyss.com/)	75.x (Windows and Android)	Web API access with JavaScript for asynchronous requests
Mozilla Firefox (https://www.mozilla.org/en-US/firefox/)	64.x (long-term support)	
Microsoft Internet Explorer	11.x (Windows 7, Windows 10)	
Microsoft Edge	44.x Windows 10	
Apple Safari	12.x iOS	
Opera	58.x	
SIMATIC HMI Panels	2	
Microsoft C# (https://docs.microsoft.com/en-us/dotnet/api/system.net.webrequest?view=netframework-4.7.2) with WebRequest class and Json.Net library (https://www.newtonsoft.com/json)	.Net Framework 4.7	Web API access for pure HTTP requests and Json.Net for generating and parsing content
GNU Wget (https://www.gnu.org/software/wget/)	1.20 Windows	Web API access for pure HTTP requests, e.g. for automatic archiving of data logs.
cURL (https://curl.haxx.se/)	7.63.x Windows	
Python (https://www.python.org/downloads/)	3.10	
Microsoft PowerShell	5.0	Web API access for pure HTTP requests with Invoke WebRequest and ConvertTo-Json/ConvertFrom-Json for generating and parsing content
WebserverApi Client Library for .NET (https://github.com/siemens/simatic-s7-webserver-api)	1.0.1	Web API access for pure HTTP requests in C#. The library is also available as a NuGet package at the following address (https://www.nuget.org/packages/Siemens.Simatic.S7.Webserver.API).

* Not included in the scope of delivery of the product described here

Github

You can also find examples of using API methods in practice on Github at the following repository (https://github.com/siemens/simatic-s7-webserver-api).

6.2 Web API integration

In the following section you will find examples of how to integrate the Web API into your application.

Structure of an HTTP request and HTTP response

The following section shows the required structure of an HTTP request and HTTP response for successfully making a Web API request.

```
POST /api/jsonrpc HTTP/1.1
Host: 192.168.3.14
Content type: application/json
Content length: 92

[{"jsonrpc":"2.0","method":"Api.Login","params":{"user":"User1","password":"SecurePassword"},"id":999}]

HTTP/1.1 200 OK
Content type: application/json
Cache-Control: no-cache
Pragma: no-cache
Expires: 0
Access-Control-Allow Origin: *
Access-Control-Allow Headers: Content-Type,X-Auth Token
Access-Control-Allow Methods: POST,OPTIONS
transfer-encoding: chunked
date: Tue, 23 Apr 2019 17:50:31 GMT

[{"jsonrpc":"2.0","id":999,"result":{"token":"Sy8pe3VNv86rTMldzFBsY-zmw12Lg"}}]
```

Web API examples

The following section contains examples of how you can use the described methods in the Web API.

The examples use HTML, JSON and JQuery library for asynchronous requests.

NOTE

Information used in the examples

Note that the names of the methods, parameters and the JavaScript code are specified without liability and can deviate from the current API methods.

Example 1

Example 1 shows a code section with which a session can be maintained. For this purpose, a one-time ping request is sent using the Api.Ping method. If the intervals at which the ping request is sent are less than the timeout of 2 minutes, the user remains permanently logged in.

A maintained session lends itself to scenarios such as monitoring and control tasks.

```
$.post({
url:"https://192.168.2.132/api/jsonrpc",
headers:{
 'X-Auth-Token':"Sy8pe3VNv86rTMldzFBsYzmw12Lg"
},
data:JSON.stringify({"jsonrpc":"2.0", method:"Api.Ping", "id":1}),
success:function(data){ console.log(data); }
dataType:"text",
contentType:"application/json"
});
```

NOTE

Authentication token

To extend the session, you must send the authentication token (X-Auth-Token) as an HTTP header to the CPU.

When you call the Api.Ping method without a token, the session is not extended because the CPU cannot assign a token to the user.

Example 3 shows an example of a token in the HTTP request.

In the example, the selected user has the necessary authorizations. The methods after the login request were successfully carried out, as the following result shows.

```
{"jsonrpc":"2.0","id":1,"result":"ZWlmbnJwZmplb3Nwd2l1Y3N3dWE="}
```

Example 2

Example 2 shows a client that wants to log in to the CPU with JavaScript and calls several methods within an HTTP request using a bulk request.

```
$.post({
  url:"https://192.168.2.132/api/jsonrpc",
  data:JSON.stringify([
    {"jsonrpc":"2.0", "id":1, method:"Api.Login",
      params:{user:"Admin",password:"12345"} },
    {"jsonrpc":"2.0", "id":2, method:"Api.GetPermissions" },
    {"jsonrpc":"2.0", "id":3, method:"Api.Browse" }]),
  success: function(data){ console.log(data); },
  dataType: "text",
  contentType: "application/json"});
```

The following section shows an example of a bulk request response. The selected user has the necessary authorizations. The methods after the login request were successfully carried out with the authorizations of the authenticated user.

```
[
"jsonrpc": "2.0", "id": 2, "result":[
{ "name": "Api.Browse" },
{ "name": "Api.Login" },
{ "name": "Api.Logout" },
{ "name": "Api.GetPermissions" },
{ "name": "PlcProgram.Read" },
{ "name": "PlcProgram.Write" }
]
```

Example 3

Example 3 shows a bulk request for read and write access to a stack of tags in a single HTTP request. This procedure is recommended for bulk requests, as it is more efficient than a series of single accesses and therefore places less load on the CPU.

```
$.post({
  url:"https://192.168.2.132/api/jsonrpc",
  data:JSON.stringify([
    {"jsonrpc":"2.0", "id":1, method:"PlcProgram.Read"},
    {"jsonrpc":"2.0", "id":2, method:"PlcProgram.Read",
      params:{"var":"\"MyDB\".InvalidField"}},
    {"jsonrpc":"2.0", "id":3, method:"PlcProgram.Read",
      params:{"var":"MyDB.MyDate"} },
    {"jsonrpc":"2.0", "id":4, method:"PlcProgram.Write",
      params:{"var":"\"BoilerControl\".TempSetPoint", value:9001}
}]),
  success: function(data){ console.log(data); },
  dataType: "text",
  contentType: "application/json",
  headers: {
    "X-Auth-Token":"d29kamV3cGxtdm5keHNhcXd1aXJ0empkZXN3cQ=="}});
```

The bulk request contains an invalid tag with an error message providing information about this. All other methods were successfully carried out, as the following result shows.

```
[
{"jsonrpc":"2.0","id":1,"result":{"value":42}},
{"jsonrpc":"2.0","id":2,"error":{"code":201,"message":"Invalid
 address"}},
{"jsonrpc":"2.0","id":3,"result":{"value":"1990-01-01"}},
```

```
{"jsonrpc":"2.0","id":4,"result":true}
]
```

Framework for simple provision of web pages

The application example "Provision of a framework for web page upload (https://support.industry.siemens.com/cs/ww/en/view/109814059)" provides an instruction for simple provision of web pages on the basis of the Web API directly onto the CPU. The application example provides you with a simple possibility to use the functionalities of the Web API without having to take HTTP requests into account.

6.3 Web API sessions

Timeout for Web API sessions

NOTE

If a Web API call is not made within a session before 120 seconds have elapsed, the CPU terminates the session with a logout event. A timeout reset is initiated by every successful action of the user in which a toke is supplied.

Call the Api.GetPermission or Api.Ping method cyclically within the timeout grid to ensure that:

- Your session remains active
- Your authorizations for the call of other methods remain active

Limitations for Web API sessions

The CPU limits the number of active sessions. The following table shows the limitations, meaning the available resources depending on the CPU used.

CPUs	Limitation
1510 to 1513	50
1514, 1515, 1516 and 1504D TF	100
1517, 1518, 1516T and 1507D TF	200

Limitation of the active Web API sessions

If you request another authentication token and none are available, the request is rejected.

Each call of Api.Login means a new authentication token. This applies for every user, also for activated "Anonymous" user.

Change as of firmware version V4.0: In contrast to users with password, only one resource is still occupied for the "Anonymous" user in the CPU, instead of <n> resources, if "Anonymous" authenticates themselves <n> times in parallel. This means that Anonymous users no longer occupy the maximum session resources of the CPU.

NOTE

The authorization on the web server as an "Anonymous" user is only foreseen for test purposes, commissioning, etc. In other words, when the system is not in productive operation. In this case, you have to ensure the security of the plant through other organizational measures, e.g. spatial protection. This user is described in more details in the section User management (Page 41).

Changes to CPU user administration

The following applies to configured CPUs with firmware version ≤ V3.0: If the configuration of the CPU user management changes (by downloading the HW configuration in the TIA Portal), e.g. password changed or user removed, the CPU terminates all sessions with a logout event.

The following applies to configured CPUs with firmware version ≥ V3.1: If a user has been authenticated and the project is subsequently loaded into the CPU, deleted or deactivated users are logged out. If only the password or role of the user changes, the user remains authenticated.

Security events

The following applies to configured CPUs with firmware version ≤ V3.0: The CPU generates a security event for successful and failed logins. The CPU enters this security event in the diagnostics buffer.

For configured CPUs with firmware version ≥ V3.1, security events are logged in the CPU's internal syslog buffer.

For more information on Syslog messages, refer to the S7-1500, ET 200MP System Manual (https://support.industry.siemens.com/cs/ww/en/view/59191792).

Authentication token

You must be able to limit the access to the functions of the CPU in order to ensure both the security as well as the protection of your process. This functionality was realized in the previous web server by specifying user accounts and authorizations. No authorization or several authorizations can be assigned to each user. When the user executes functions on the CPU, the CPU checks whether the user disposes of the corresponding authorizations before the requested action is carried out.

The authentication token is the result of a successful login using the Api.Login method.

This token must be transferred to the Web API at all the subsequent API requests after a successful login as an HTTP header "X-Auth-Token".

In the following example, the X-Auth-Token is passed on as an HTTP header to the request:

```
$.post({
```

```
url:"https://192.168.2.132/api/jsonrpc",
data:JSON.stringify({jsonrpc:"2.0",id:1,
  method:"RequestChangeOperatingMode",params:{mode:"run"}}),
success: function(data){ console.log(data); },
dataType: "text",
contentType: "application/json",
headers: {
  "X-Auth-Token":"VGhpc0lzU2hvcnRUb1ByZXZlbnRXcmFwcGluZw=="}});
```

You can find a description of the method in the section Api.Login (Page 85).

6.4 Web API basic functions

The following section gives an overview of all Web API basic functions with extracts from the corresponding HTML code.

NOTE

Files which contain Web API methods must be encoded and stored in the UTF-8 character encoding.

For detailed examples of an integration of the Web API into your web application, refer to the section Web API integration (Page 79).

6.4.1 Api.Login

This method checks the login data of the user and on successful verification opens a new Web API session. The method requests the name and the password of the user in plain text as proof of authorization. The user name and the password are encrypted before they are transferred to the server.

No authorization is required to call the Api.Login method.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-1 Api_Login_Request (object)

Name	Required	Data type	Description
mode	No	string	Login mode. You can use all the modes supported by the Api.GetAuthenticationMode method, see Api.GetAuthenticationMode (Page 94). If no parameter is used, the query is assumed to be "local".
user	Yes	string	The user name. The parameter must be specified in the following modes: "static", "disabled", "local" and "umc".
password	Yes	string	The current password in plain text, without encryption. The parameter must be specified in the following modes: "static", "disabled", "local" and "umc".
include_web_application_cookie	No	bool	This parameter specifies: • Whether a "web_application_cookie" cookie was generated for access to protected web applications • Whether you want to return the cookie with the response to the successful login

Example 1

The following example shows the parameters required to call an Anonymous user.

```
{
  "user": "Anonymous",
  "password": ""
}
```

Example 2

The following example shows the parameters required to call a regular user.

```
{
  "user": "user_1",
  "password": "1234"
}
```

Example 3

The following example shows the parameters required to call an Anonymous user with local user management.

```
{
  "mode": "local",
  "user": "user_1",
  "password": "1234"
}
```

Example 4

The following example shows the parameters required to call an Anonymous user with central user management.

```
{
  "mode": "umc",
  "user": "user_1",
  "password": "1234"
}
```

Token

The token comprises a 28-byte string. The token is transferred in encrypted form.

For every additional request which requires authentication, you have to specify the assigned token in the HTTP request header. If further communication no longer takes place in the meantime, the token becomes invalid after maximum 2.5 minutes. Each new request within a session extends the validity of the token by another 2 to 2.5 minutes, calculated from the completion of the request processing by the server.

The token is not required for methods that do not require authentication. However, you can still enter the token. If the token is passed when a method is called, the timeout of the corresponding session is reset.

When you call, for example, the Api.Ping method without a token, the session is not extended because the CPU cannot assign a token to the user.

The following methods, amongst others, work with and without tokens:

- Api.Browse
- Api.Ping
- Api.GetPermissions

Receiving a token for a passwordless user account without password

If a user wants to use a user account without password to access the Web API, they must authenticate with the API method Api.Login using the user name "Everybody"/"Anonymous" and an empty password ("").

Even if only the central user management was activated, an Anonymous user can be present on the CPU, in as far as it was activated. In this case the GetAuthenticationMode method only returns "umc", but not "local". Nevertheless, a login with mode=local or without mode parameter is possible.

The name of the specific user depends on the firmware version of the CPU:

- For projects with a firmware version ≤ V3.0: Static user management with the user "Everybody"
- For projects with a firmware version ≥ V3.1: Local user management with "Anonymous" user
- For projects with a firmware version ≥ V4.0: Local or central user management with "Anonymous" user

The "Anonymous" user can be deactivated in STEP 7 and is then not available on the CPU. Non-activated users are not downloaded to the CPU. This means that a call of the API method Api.Login with the user name "Anonymous" fails if the user was deactivated in STEP 7.

NOTE

User "Everybody"/"Anonymous"

Note that, unlike TIA Portal, the Web API only accepts the English notation "Everybody" or "Anonymous".

Response structure

The following tables show the structure of server responses to successful requests.

Table 6-2 Api_Login_Response (object)

Name	Required	Data type	Description
token	Yes	string	The token indicates that its holder has successfully authenticated themselves at the Web API. The returned token must be passed via the HTTP request header "X-Auth-Token" in subsequent Web API requirements.
web_application-_cookie	No	string	Only present if "include_web_application_cookie" is "true". The holder of the token has successfully authenticated themselves with the Web API and has authorization to access protected web applications.
password_expiration	No	object of type `Api_Login_PasswordExpiration_Response`	This object contains information on the expiration of the password, if: • The "local" or "umc" authentication mode is used for the CPU and • The password policy is activated on the CPU
runtime_timeout	No	string	ISO 8601 time span as string Time span of inactivity in seconds after which a client application is to perform a logout using the API method Api.Logout.

Table 6-3 Api_Login_PasswordExpiration_Response (object)

Name	Required	Data type	Description
timestamp	Yes	string	ISO8601 time stamp as a string in Coordinated Universal Time (UTC) Indicates when the user password expires. The accuracy must be specified in seconds.
warning	Yes	bool	This parameter specifies whether the warning threshold was reached before the password expired.

Example 1

The example below shows a successful login for a user without expiration of the password (either static user management or password expiration deactivated):

```
{
  "token": "TXlMdWdnYWdlSGFzVGhlU2FtZSE="
}
```

Example 2

The following example shows a successful login for a user with password expiration:

```
{
  "token": "TXlMdWdnYWdlSGFzVGhlU2FtZSE=",
  "password_expiration":
  {
     "timestamp": "2023-11-05T18:25:43Z",
     "warning": true
  }
}
```

Example 3

The following example shows a successful login for a user with password expiration and runtime_timeout of 30 minutes:

```
{
  "token": "TXlMdWdnYWdlSGFzVGhlU2FtZSE=",
  "runtime_timeout": "PT30M",
  "password_expiration":
  {
     "timestamp": "2023-11-05T18:25:43Z",
     "warning": true
  }
}
```

Possible error messages

The following table shows possible error messages of the Api.Login method.

Error code	Error message	Meaning
4	No resources	The system does not have the required resources to carry out this request. Perform the request again as soon as enough resources are available again.
6	Not accepted	The authentication cannot be performed because the mode is not supported by the CPU.
100	Login failed	The user name and/or password are not permissible. Assign a permissible user name and a permissible password. Another reason why the login failed may be an active brute force attack.
101	Already authenticated	The current authentication token is already authenticated. Use `Api.Logout` before you authenticate yourself again.
102	Login Failed - Password expired	The password of the user account has expired. The user must change the password in order to be able to successfully authenticate again.
105	Login Failed - Infrastructure Error	The authentication was not successful, for example because the server cannot be reached.

6.4.2 Api.Logout

The Api.Logout method removes the token from the list of active Web API sessions and ends the session.

No authorization is required to call the Api.Logout method.

The Api.Logout method returns the status of whether the logout was successful or not. For security reasons, however, the method always returns the Boolean value "true" even if the token is invalid.

6.4.3 Api.GetPermissions

After a successful login, this method returns a list of actions for whose execution the user is authorized.

No authorization is required to call the Api.GetPermissions method.

Response structure

The following table shows you the structure of server responses to successful requests:

Table 6-4 Api_GetPermissions_Response (array of objects)

Name	Required	Data type	Description
name	Yes	string	Name of authorization

Example

The following example shows the actions for which the user is authorized.

```
[
  { "name": "read_value" },
  { "name": "change_operating_mode" }
]
```

Checkable authorizations

You can use the Web API to check the authorizations for the following functions.

Action	User authorization	As of the firmware version of the CPU
flash_leds	Identify device	V2.8
acknowledge_alarms	Acknowledge alarms	V2.8
read_value*	Read process data from the CPU.	V2.8
write_value	Write process data to the CPU.	V2.8
read_diagnostics*	Query diagnostics data from the CPU without being permitted to change data.	V2.8
change_operating_mode*	Change the operating mode.	V2.8
read_file*	Read the contents of files on the CPU.	V2.8
write_file*	Change the contents of files and folders on the CPU.	V2.8
backup_plc	Back up the CPU configuration.	V2.8
restore_plc	Restore configuration of the CPU.	V2.8
failsafe_admin	Make fail-safe changes on the CPU.	V2.8
open_user_pages	Call user-defined pages on the CPU.	V2.8
manage_user_pages	Change user-defined pages on the CPU.	V2.9
update_firmware	Perform firmware update	V3.0
read_watch_table_value	Read the value of a tag in the watch table	V3.1
write_watch_table_value	Write the value of a tag in the watch table	V3.1
read_syslog*	Read the SysLog buffers of the CPU	V3.1
change_time_settings	Change the system time settings of the CPU	V3.1
change_webserver_default_page	Change the default page of the Web server	V3.1
download_service_data	Load the service data of the CPU	V3.1
change_webserver_response_headers	Change the response header of the Web server	V4.0
manage_system_pages	Update or configure system pages of the web server	V4.0

* Also available for the R/H-CPU of the redundant S7-1500R/H system (as of firmware version V3.1).

For projects with firmware version ≤ V3.0, the Web API checks the authorization based on the rights and passwords assigned in STEP 7 in the Inspector window in the "Web server > User management" area.

For projects with firmware version ≥ V3.1, the Web API checks the authorization based on the rights and passwords assigned in STEP 7 in the project navigation in the "Security settings > Users and roles" area.

A description of the user management can be found in section "Configuring the Web server (Page 22)".

6.4.4 Api.ChangePassword

You can change the password for a user account with this method.

Recommendation: Before changing a password, read the password policy from the CPU using the Api.GetPasswordPolicy API method. If the new password does not conform to the password policy of the CPU, a corresponding error message is returned.

No prior authorizations are required to call the Api.ChangePassword method, but you must enter the current password for this call.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-5 Api_ChangePassword_Request (object)

Name	Required	Data type	Description
mode	No	string	Password change mode. Possible values: • local The values are described in the Api.GetAuthenticationMode method. If no parameter is entered, "local" is used.
username	Yes	string	The user name for which the password is changed
password	Yes	string	The current password of the specified user
new_password	Yes	string	The new password. Select a secure password in order to exclude negative effects on the CPU.

Example

In the following example, the password is changed for the user account "HappyUser".

```
{
  "username": "HappyUser",
  "password": "mycurrentpassword",
  "new_password": "mynewpassword"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

The following table shows possible error messages of the Api.ChangePassword method.

Error code	Error message	Meaning
4	No resources	No resources for the password. Only a parallel resource is available for the password change.
5	System is read-only	The memory card is write-protected. Therefore, the password cannot be changed.
6	Not accepted	The password change is not performed because a CPU was configured a with firmware version < V3.1 or the selected mode does not support the function. The method can only be used with CPUs as of firmware version V3.1.
100	Login failed	The user name and password combination is invalid. Assign a permissible user name and a permissible password. Another reason why the login failed may be an active brute force attack.
103	New password does not follow password policy	The provided new password does not match with the required password policy. Assign a password conforming to the password policy. The Api.GetPasswordPolicy method provides you with the password policy of the CPU, if the CPU is in "local" authentication mode.
104	New password matches former password	The new password is identical with the previous password. Assign a different password. Note that the CPU does not store a password history. The comparison is therefore only performed between the new and previous password.

6.4.5 Api.GetPasswordPolicy

This method provides you with the password expiry policy of the CPU when the CPU is in the authentication mode "local" or "umc".

Passwords must fulfill specific criteria to be secure. The Api.GetPasswordPolicy method provides you with the password policy of the CPU. The password policy is a global setting in the STEP 7 project and applies for all users of the Web server. The method does not contain any information on the expiration of the password. Any user, including unauthenticated users ("AnonymousAnonymous"), can call this API method.

No authorization is required to call the Api.GetPasswordPolicy method.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-6 Api_GetPasswordPolicy_Request (object)

Name	Required	Data type	Description
mode	No	string	Password change mode. Possible values: • local The values are described in the Api.GetAuthenticationMode (Page 94) method. If no parameter is entered, "local" is used.

Response structure

The following table shows you the structure of server responses to successful requests.

Table 6-7 Api_GetPasswordPolicy_Response (object)

Name	Required	Data type	Description
password_policy	Yes	object	The object represents the current password policy of the CPU.
min_password_length	Yes	number	The minimum length of the password in UTF-8 character encoding. Value range: 8 to 255
max_password_length	Yes	number	The maximum length of the password in UTF-8 character encoding.
min_digits	Yes	number	The minimum number of numerals (0 to 9) within the password. Value range: 0 to 255
min_special_characters	Yes	number	The minimum number of special characters within the password. Special characters are: !#$%&()*+,-./:;<=>?@[\]_{|}~^ Value range: 0 to 255
requires_uppercase_characters	Yes	bool	The password contains at least one uppercase character A to Z. Values: true or false
requires_lowercase_characters	Yes	bool	The password contains at least one lowercase character a to z. Values: true or false

Example

The following example shows a CPU project with a password policy.

```
{
 "password_policy":
  {
"min_password_length": 8,
"max_password_length ": 120,
"min_digits": 2,
   "min_special_characters": 1,
   "requires_uppercase_characters": true,
   "requires_lowercase_characters": true
  }
}
```

Possible error messages

The following table shows possible error messages of the Api.GetPasswordPolicy method.

Error code	Error message	Meaning
6	Not accepted	The password change is not performed because a CPU was configured a with firmware version < V3.1 or the selected mode does not support the function. The method can only be used with CPUs as of firmware version V3.1.

6.4.6 Api.GetAuthenticationMode

You can read the current authentication mode of the CPU with this method.

No authorization is required to call the Api.GetAuthenticationMode method.

Possible authentication modes

The following authentication modes are available to you with the Web server.

Table 6-8 Authentication modes

Mode	String	Meaning	Supported as of the configured FW version of the CPU
Static user management	`static`	In this mode, changes are only possible by downloading the hardware configuration from the CPU. Passwords cannot be changed during runtime of the user program.	V2.8 to V3.0
Access control deactivated	`disabled`	No user management and authentication is possible in this mode. Only the "Anonymous" user is available. This specific user has full access/rights to the CPU and Web server functionality. As with every other user the Api.Login method must be called. The returned X-Auth Token allows the full access right to all API functionalities.	V3.1
Local user management	`local`	In this mode, the authorizations and user roles continue to be configured as part of the STEP 7 project and a password change is possible via the Web API.	V3.1
Central user management (via user name and password)	`umc`	In this mode the user management is controlled by the central server (UMC server). The user roles are stored in the CPU: Changing the password is not possible via the Web API. In this mode the CPU checks the login data against the UMC server.	V4.0

Response structure

The following table shows you the structure of server responses to successful requests.

Table 6-9 Api_GetAuthenticationMode_Response (object)

Name	Required	Data type	Description
authentication_modes	Yes	array of strings	The parameter describes the current authentication mode supported by the CPU.

Example 1

In the following example, the authentication mode "local" (local user management) is read by the CPU.

```
{
  "authentication_modes": ["local"]
}
```

Example 2

In the following example, the authentication mode "local" (local user management) and umc (central user management) is read by the CPU.

```
{
  "authentication_modes": ["local", "umc"]
}
```

6.4.7 Api.GetSessionInfo

This method is used to read the information about the current API session.

No authorization is required to call the Api.GetSessionInfo method.

Response structure

The following table shows you the structure of server responses to successful requests.

Table 6-10 Api_GetSessionInfo_Response (object)

Name	Required	Data type	Description
authentication_mode	No	string	The parameter describes the authentication mode used when the user session is opened: • Static user management ("static") • Local user management ("local") • Central user management (via user name and password; "umc") No authentication mode is returned for the users "Anonymous"/"Everybody".
username	Yes	string	The parameter contains the user name of the session.
password_expiration	No	Object of type `Api_GetSessionInfo_PasswordExpiration_Response`	This object contains information on the expiration of the password, if: • The "local" or "umc" authentication mode is used for the CPU and • The password policy is activated on the CPU
runtime_timeout	No	string	ISO 8601 time span as string Time span of inactivity after which a client application is to perform a logout using the API method Api.Logout.

Table 6-11 Api_GetSessionInfo_PasswordExpiration_Response (object)

Name	Required	Data type	Description
timestamp	Yes	string	ISO8601 time stamp as a string in Coordinated Universal Time (UTC) Indicates when the user password expires. The accuracy must be specified in seconds.
warning	Yes	bool	This parameter specifies whether the warning threshold was reached before the password expired.

Example 1

The following example describes a session opened with static user management.

```
{
  "authentication_mode": "static",
  "username": "MyUser"
}
```

Example 2

The following example describes a session opened through deactivated access control. Only the "Anonymous" user is available.

```
{
  "username": "Anonymous"
}
```

Example 3

The following example describes a session opened with central user management for the user with timeout and expiry of the password.

```
{
  "authentication_mode": "umc",
  "username": "MyUser",
  "runtime_timeout": "PT30M",
  "password_expiration":
  [
     "timestamp": "2012-04-23T18:25:43Z",
     "warning": true
  }
}
```

Example 4

The following example describes a session opened with local user management for the user with timeout and expiry of the password.

```
{
  "authentication_mode": "local",
  "username": "MyUser",
  "runtime_timeout": "PT30M",
  "password_expiration":
  {
     "timestamp": "2012-04-23T18:25:43Z",
     "warning": true
  }
}
```

Example 5

The following example describes a session opened with local user management for the user without expiry of the password.

```
{
  "authentication_mode": "local",
  "username": "MyUser",
  "runtime_timeout": "PT30M"
}
```

Possible error messages

The following table shows possible error messages of the Api.GetSessionInfo method.

Error code	Error message	Meaning
6	Not accepted	The method cannot be carried out due to an authentication token.

You can find more information on the authentication token in the "Web API sessions (Page 82)".

6.4.8 Api.Browse

The Api.Browse method gives you a list of all methods that you can call via the Web API with the current firmware. This provides you with an overview of all the methods supported by the CPU.

No authorization is required to call the Api.Browse method.

Response structure

The following table shows you the structure of server responses to successful requests.

Table 6-12 Api_Browse_Response (array of objects)

Name	Required	Data type	Description
name	Yes	string	The name of the method.

Example

The following example below shows the supported methods of the CPU.

```
[
 { "name": "Api.Browse" },
 { "name": "Api.Login" },
 { "name": "Api.Logout" },
 { "name": "Api.GetPermissions" },
 { "name": "PlcProgram.Read" },
 { "name": "PlcProgram.Write" }
]
```

NOTE

Checking authorizations

The Api.Browse method does not filter the list of the available methods by the individual authorizations of users.

The list of available methods may therefore contain methods which the user may not execute without authorization.

Possible error messages

The following table shows possible error messages of the Api.Browse method.

Error code	Error message	Meaning
4	No resources	The system does not have the necessary resources to execute the Web API request. Perform the request again as soon as enough resources are available again.

6.4.9 Api.Version

With this method you request the current version number of the Web API. You can draw conclusions from the version number:

- The functions supported by the CPU version
- The hardware functional status of the CPU

This information lets you implement applications that dynamically adapt to the scope of functions offered by the contacted CPU. An application can support multiple CPU versions.

No authorization is required to call the Api.Version method.

Response structure

The version number is displayed as a floating-point number and is incremented with every release and every change of the Web API.

Example

The following example shows a possible result of calling the Api.Version method.

```
5.2801020
```

6.4.10 Api.Ping

This method outputs a unique ID for the CPU used. You can use it to determine whether the CPU can be reached. The CPU ID comprises a 28-byte string. The system assigns a new, unique CPU ID after each restart (POWER ON - POWER OFF) or warm start of the CPU. By comparing this with previously output IDs, you can also determine whether the CPU was restarted in the meantime.

No authorization is required to call the Api.Ping method.

Response structure

The method issues a string that contains 20 bytes of "random" data generated by the CPU. Encrypted with Base64 it represents a total of 28 bytes.

Example

The following example shows the output of a CPU ID:

```
"QXJlWW91UmVhbGx5U2VyaW91cz8="
```

6.4.11 Api.GetCertificateUrl

This method outputs a relative URL (`https://<IP>`) with which you can call the SSL certificate of the web server.

No authorization is required to call the Api.GetCertificateUrl method.

Response structure

The method outputs a string with a relative URL to the root directory of the CPU Web server (`https://[ip-address]`).

If the Web server has not been configured with a CA certificate generated via the global security settings, the method outputs an empty string.

Example

The following example shows the result of the Api.GetCertificateUrl method call.

```
"/MiniWebCA_Cer.crt"
```

6.4.12 Api.GetQuantityStructures

This method returns various units of structure information of the web server.

No authorization is required to call the Api.GetQuantityStructures method.

Response structure

The following table shows you the structure of server responses to successful requests.

Table 6-13 Api_GetQuantityStructures_Response (object)

Name	Required	Data type	Description
webapi_max_http_request_body_size	Yes	number	The maximum size of the HTTP request body of a Web API request in bytes
webapi_max_parallel_requests	Yes	number	The maximum number of parallel requests at the web application end point
webapi_max_parallel_user_sessions	Yes	number	The maximum number of parallel user sessions that use the web application end point

Example

The following example shows the response with the read parameters of a CPU.

```
{
  "webapi_http_request_body_size": 131072,
  "webapi_parallel_requests": 4,
  "webapi_parallel_user_sessions": 200
}
```

6.5 Ticket mechanism

With the web server as of firmware version V2.9, you can use the ticket mechanism of the Web API. The ticket mechanism is the basis for all file-based methods, such as the download of resources from the CPU.

The ticket mechanism enables you to:

- Transfer large amounts of data outside of the JSON-RPC protocol.
- Call status information, for example, to implement progress indicators in your application and respond conveniently to mode changes

NOTE

4 tickets per session

The ticket mechanism enables you to use a maximum of 4 tickets simultaneously per session.

The number of tickets that can be created per type is additionally limited. For more information, see the API methods that create a ticket:

- WebApp.CreateResource (Page 138)
- WebApp.DownloadResource (Page 142)
- Plc.CreateBackup (Page 207)
- Files.Download (Page 216)
- Files.Create (Page 218)
- PlcProgram.DownloadProfilingData (Page 162)

Principle of the ticket mechanism

Data transfers outside the JSON-RPC protocol are initiated by Web API methods, such as WebApp.DownloadResource. This method returns a specific identifier, a so-called "ticket". The ticket can be redeemed by a subsequent request to another HTTP end point. In the request, the data are exchanged with standard HTTP mechanisms.

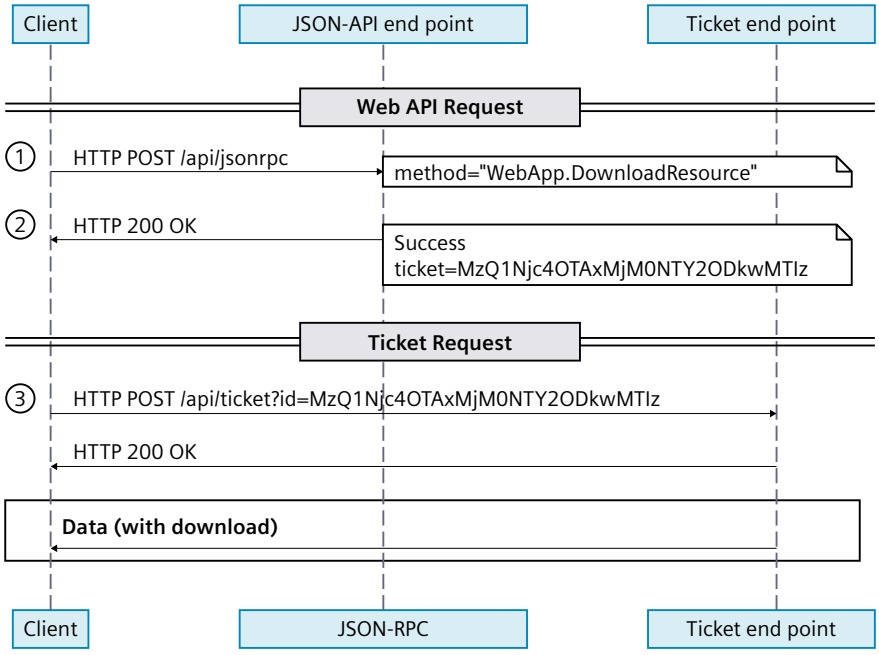


Figure 6-2 Ticket mechanism

① To request the upload or download of a resource from/to the CPU, a client sends a request to the JSON-RPC.

② After successful authorization and resource check, the client receives a valid ticket ID.

③ The client sends a request to the ticket end point with the valid ticket ID and an X-Auth-Token. The data is included here during upload to the CPU. The request is confirmed to the client with "HTTP 200 OK" or "204 NO CONTENT". The data is returned in a download from the CPU.

Ticket IDs are one-time tokens that may not be reused by the client or by the CPU after being redeemed.

NOTE

Changes in firmware version V3.0

As of firmware version V3.0, no X-Auth-Token is required for the ticket end point.

Do not share the token with third parties, because the owner of the token gets control over a ticket.

As of firmware version V3.0, GET requests for downloads are possible in addition to POST requests. For GET requests, a content disposition header is returned with the HTTP response. It contains a default file name for storage in the web browser. The browser uses the file name as the storage location. However, you can also evaluate the file name programmatically using a programming language (for example, via C#).

NOTE

Changes in firmware version V4.0

As of firmware version V4.0 file names with special characters are supported if the `filename*` attribute of the content disposition header is used.

File names for CPUs as of firmware version V4.0

As of firmware version V4.0 file names with special characters are supported if the `filename*` attribute of the content disposition header is used. In addition to the `filename` attribute, the header also contains the `filename*` attribute that is used implicitly and is interpreted in accordance with the encryption process. In the older web browsers `filename*` is not implemented. `filename` is therefore used.

Example of an HTTP content disposition header

```
Content disposition: attachment;
filename*=UTF-8''%e9%a1%b9%e7%9b%ae1.zap19;
filename="e9a1b9e79bae1.zap19"
```

Ticket end point

The ticket end point is accessible via GET (for downloads only), POST and OPTIONS requests to the following URL:

`https://[ip]/api/ticket?id=<ticket-id>`

The ticket mechanism alone is not enough to execute a file action. To do this, you must call a specific method. In the web server of the firmware version V2.9, you use the ticket mechanism for the following methods:

- WebApp.CreateResource (Page 138)
- WebApp.DownloadResource (Page 142)

Procedure to execute a file action

To create a resource or download one from the CPU, follow these steps:

1. Call the corresponding method with the necessary method parameters, for example, WebApp.DownloadResource. The method returns a character string that includes a valid ticket ID.
2. Call the ticket end point via POST request with the returned ticket ID. The method will then execute the download, for example.

For more information, see the paragraph "Principle of the ticket mechanism".

NOTE

In some cases, programming languages perform preprocessing of text files before the upload. For example, a UTF-8 BOM-encoded (Byte Order Mark) file in Javascript is converted to a UTF8 file in advance.

Examples

Below you will find two examples for further processing of the tickets. The examples use HTML, Json and JQuery library.

WebApp.CreateResource (upload a resource):

```
<td><input id="fileForticketCustomerExampleUpload" type="file"
onchange="fReadFile()" /></td>
var fileReader = new FileReader();
function fReadFile() {
   fileReader = new FileReader();
   fileReader.readAsText(this.files[0]);
}
$("#ApiUploadticketCustomerExample").click(function () {
     var ticketId = $("#iApiUploadticketIdCustomerExample").val();
     var content = fileReader.result; // e.g. <!DOCTYPE
html><html><head>...
     $.post({
       url: "https://" + window.location.hostname +
"/api/ticket?id=" + ticketId,
       headers: { "X-Auth-Token": token, "Content-Type":
"application/octet-stream" },
       contentType: "application/json",
       data: content,
       // ticketing: status = 204: No content (=> no data) upload
has finished successfully:
       success: function (data, textStatus, jqXHR) { if
(jqXHR.status == 204) {
$("#ApiUploadticketCustomerExampleRes").text("true"); } },
//print error to the document
       error: function (jqXHR, textStatus, errorThrown) {
$("#ApiUploadticketCustomerExampleRes").html("<tr><td>code:
</td><td>" + jqXHR + "</td></tr><tr><td>textStatus:</td><td>" +
textStatus + "</td></tr><tr><td>error:</td><td>" + errorThrown +
"</td></tr>"); }
     });
   });
```

WebApp.DownloadResource (download a resource):

```
// Function to save data in a file - will be stored in the default
download folder
function saveDataInFile(data, filename, type) {
   var file = new Blob([data], { type: type });
   if (window.navigator.msSaveOrOpenBlob) // IE10+
     window.navigator.msSaveOrOpenBlob(file, filename);
   else { // Others
     var a = document.createElement("a"),
       url = URL.createObjectURL(file);
     a.href = url;
     a.download = filename;
```

```
      document.body.appendChild(a);
      a.click();
      setTimeout(function () {
        document.body.removeChild(a);
        window.URL.revokeObjectURL(url);
      }, 0);
    }
}
 $("#ApiDownloadTicketCustomerExample").click(function () {
      var ticketId = $("#iApiDownloadTicketIdCustomerExample").val();
      var filename =
$("#iApiDownloadTicketCustomerExampleFileName").val();  //e.g. index
      var type =
$("#iApiDownloadTicketCustomerExampleFileType").val();    //e.g.
text/html
      $.post({
        url: "https://" + window.location.hostname +
"/api/ticket?id=" + ticketId,
        headers: { "X-Auth-Token": token, "Content-Type":
"application/octet-stream" },
        contentType: "application/json",
        // ticketing: status = 200: download has finished
successfully:
        success: function (data, textStatus, jqXHR) { if
(jqXHR.status == 200) { download(data, filename, type); } },
//print error to document
        error: function (jqXHR, textStatus, errorThrown) {
$("#ApiDownloadTicketCustomerExampleRes").html("<tr><td>code:
</td><td>" + jqXHR + "</td></tr><tr><td>textStatus:</td><td>" +
textStatus + "</td></tr><tr><td>error:</td><td>" + errorThrown +
"</td></tr>"); }
      });
    });
```

Ticket methods

Two methods are available for handling your tickets as an authenticated user:

- You can use the Api.BrowseTickets (Page 106) method to find out which tickets are currently active for you and read out the status for the active tickets in each case.
- After the action, you can delete your ticket with the Api.CloseTicket (Page 108) method to clean the application with regard to active tickets.

6.5.1 Api.BrowseTickets

This method calls the status of all active tickets of a logged-in user. Alternatively, the status of a specific ticket can be called. This will provide information on the ticket status. You can also call this method during an active file action to retrieve the current status. Depending on the file action, additional information is provided via Api.BrowseTickets.

No authorization is required to call the Api.BrowseTickets method, but a valid session token. You can find information on the session token in the section Api.Login (Page 85).

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-14 Api_BrowseTickets_Request (object)

Name	Required	Data type	Description
id	No	string	The ticket ID that was returned by an API method for use by the ticket system. If the parameter is not specified, then the response structure returns all valid tickets of the user.

Example

In the following example, a ticket ID is transferred as parameter.

```
{
  "id": "U2VyaW91c2x5LCBTdG9wIGl0ISE6"
}
```

Response structure

The following table shows you the structure of the server response to a successful request.

Table 6-15 Api_BrowseTickets_Response (object)

Name	Required	Data type	Description
max_tickets	yes	number	Maximum number of tickets for one session (4)
tickets	Yes	array of Api_ BrowseTickets_ Ticket_ Response	Ticket ID
id	yes	string	Ticket ID
date_created	yes	string	ISO8601 time stamp as string. Time of the ticket creation based on the CPU time

Name	Required	Data type	Description
provider	yes	string	Name of the method that has created the ticket, for example, WebApp.DownloadResource
state	yes	string	Current ticket status. The following alternatives are possible: "created", "active", "", "completed" or "failed"
data	No	object	Additional ticket data: Some ticket-based methods offer users additional information. This additional information is provided via "data" and described in the sections of the respective API methods.

Example

The following example shows a response of the CPU.

```
{
  "max_tickets": 4,
  "tickets":
  [
    {
      "id": "U2VyaW91c2x5LCBTdG9wIGl0ISE6",
      "date_created": "2021-01-15T08:00:00-05:00",
      "provider": "WebApp.DownloadResource",
      "state": "active"
    }
  ]
}
```

NOTE

Additional ticket data (as of TIA Portal V18)

As of TIA Portal V18, each ticket is extended by the entry "bytes_processed" as part of the "data" object. This entry specifies how many bytes have been transferred when downloading or uploading a ticket until the Api.BrowseTicket method is called.

Possible error messages

The following table shows possible error messages of the Api.BrowseTickets method.

Error code	Error message	Meaning
400	Not Found	The returned ticket ID does not exist in the ticket list of the user or does not match the assigned session token.

6.5.2 Api.CloseTicket

You use this method to delete a ticket provided by the system that is assigned to the current user session.

No authorization is required to call the Api.CloseTicket method, but a valid session token. You can find information on the session token in the section Api.Login (Page 85).

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-16 Api_CloseTicket_Request (object)

Name	Required	Data type	Description
id	yes	string	The ticket ID that was returned by an API method for use by the ticket system.

Example

In the following example, a ticket ID is transferred as parameter.

```
{
  "id": "U2VyaW91c2x5LCBTdG9wIGl0ISE6"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the Api.CloseTicket method.

Error code	Error message	Meaning
400	Not Found	The returned ticket ID does not exist in the ticket list of the user. The returned ticket ID or does not match the assigned session token.

6.6 User-configurable HTTP response headers

6.6.1 Reading and writing user-configurable HTTP response headers

User-configurable HTTP response headers

With the introduction of the SIMATIC HMI Unified Comfort Panels, the HMI Panels are also based on the web technology. It is now possible to access the contents of a Unified Comfort Panels via a web browser. The BrowserControl of an HMI-Panel loads web pages into an HTML frame.

To prevent clickjacking, Siemens products by default block the loading of web pages or applications in HTML frames. The option `sameorigin` prevents the loading of the web page in a frame. This also applies to the SIMATIC HMI Unified Comfort Panels.

The configuration and return of the `frame-ancestors` response header by the CPU web server informs a web server into which frame the web page may be loaded. Therefore you must be able to configure the response header of the CPU web server to define a list of trustworthy addresses into which the web pages can be embedded.

For CPUs as of firmware version V4.0 you can configure headers for web applications that are to be returned as part of an HTTP response. You can use response headers for different application cases, for example to increase the security of web pages, information purposes and debugging. You can configure these headers via the API.

The headers are applied to the web application endpoint and thus to all available web applications, including the system web pages. The previous web pages are not included.

Example

You have configured the CPU web server on the address 192.168.0.1 and one on the address 192.168.0.2.

Trusted clients are not configured.

If you now want to embed web pages from 192.168.0.2 into a frame on the web pages of 192.168.0.1, this does not function automatically because the web pages of 192.168.0.2 are defined as `Content-Security-Policy: sameorigin` and can therefore not be loaded into external frames.

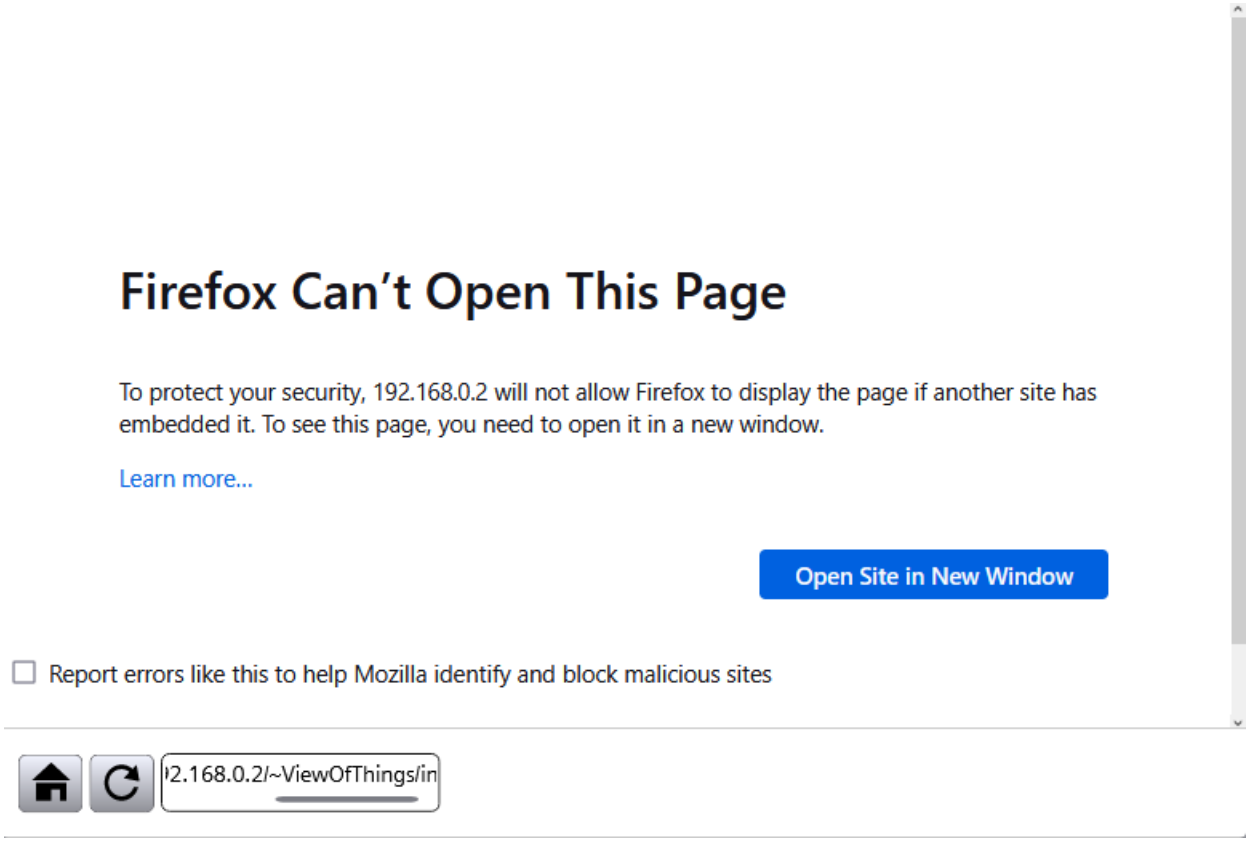


Figure 6-3 The web page cannot be loaded into the frame.

If you now configure a header `Content-Security-Policy: frame-ancestors` on the web server on the address 192.168.0.2 as follows, 192.168.0.1 is marked as trustworthy.

```
{
  "jsonrpc": "2.0",
  "id": 6,
  "result": {
    "configured_headers": [
      {
        "pattern": "~**/*",
        "header": "Content-Security-Policy: frame-ancestors 'self'
https://192.168.0.1"
      }
    ]
```

```
    "allowed_headers": [
      {
        "pattern": "~**/*",
        "key": "Content-Security-Policy"
      }
    ]
  }
}
```

After you have performed a refresh, the web page of 192.168.0.2 can be called in the frame of the web pages of 192.168.0.1. The configured headers are used on the web application end point.

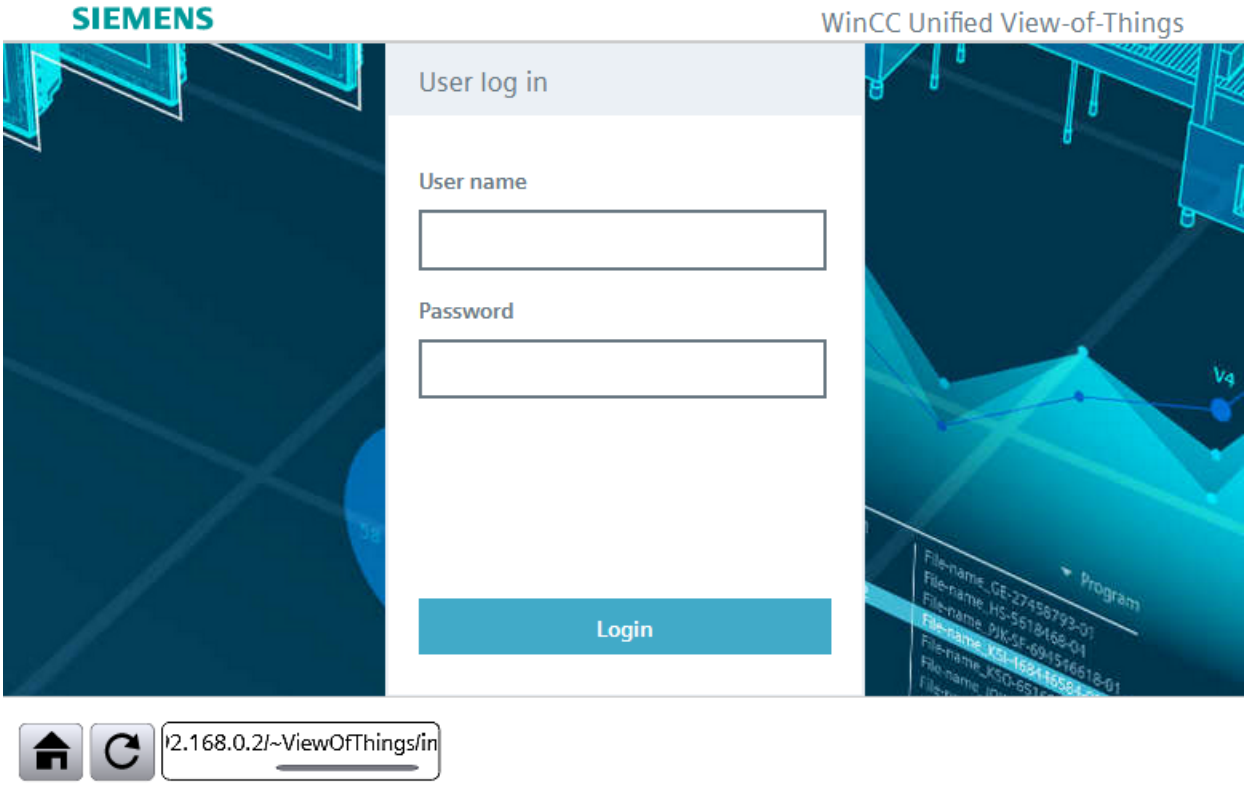


Figure 6-4 The web page can be loaded into the frame.

6.6.2 WebServer.ReadResponseHeaders

With this method, you read the list of configured HTTP response headers, as well as the list of HTTP response headers which you can configure.

To call the WebServer.ReadResponseHeaders method, you require "change_webserver_response_headers" authorization.

Response structure

The following tables show the structure of server responses to successful requests.

Table 6-17 WebServer_ReadResponseHeaders_Response (object)

Name	Required	Data type	Description
configured_headers	Yes	Array of `WebServer_ReadResponseHeaders_Configured_Response`	An array of objects. Each object represents an HTTP header that you have configured.
allowed_headers	Yes	Array of `WebServer_ReadResponseHeaders_Allowed_Response`	An array of objects. Each object represents an HTTP header that you can configure.

Table 6-18 WebServer_ReadResponseHeaders_Configured_Response (object)

Name	Required	Data type	Description
pattern	Yes	string	The sample describes for which URLs this HTTP response header is returned. The parameter must always contain the value "/~**/*". This means that this header can be applied to all URLs, which begin with "~".
header	Yes	string	The HTTP response header that is added to the HTTP response

Table 6-19 WebServer_ReadResponseHeaders_Allowed_Response (object)

Name	Required	Data type	Description
pattern	Yes	string	The sample for which the header can be used.
key	Yes	string	A pre-defined or user-defined key. Only the following value is possible: • "Content-Security-Policy"

Example 1

The following example shows a response header with permitted list.

```
{
  "configured_headers": [
  {
      "pattern": "~**/*",
      "header": "Content-Security-Policy: frame-ancestors
*.somesite.com;"
    }
  ],
```

```
  "allowed_headers": [
  {
      "pattern": "~**/*",
      "key": "Content-Security-Policy"
    }
  ]
}
```

Example 2

The following example shows a non-configured response header.

```
{
  "configured_headers": [],
  "allowed_headers": [
  {
    "pattern": "~**/*",
    "key": "Content-Security-Policy"
  }
}
```

Possible error messages

The following table shows possible error messages of the WebServer.ReadResponseHeaders method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.

6.6.3 WebServer.ChangeResponseHeaders

With this method, you can overwrite and configure a set of user-defined HTTP response headers. To ensure the correct sequence of the HTTP headers and simplify the use, all HTTP response headers configured by you are specified in a API call.

Currently only headers with the key "Content-Security-Policy" are supported.

To call the WebServer.ChangeResponseHeaders method, you require "change_webserver_response_headers" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-20 WebServer_ChangeResponseHeaders_Request (object)

Name	Required	Data type	Description
pattern	Yes	string	The sample for which the header must be output. The parameter must always have the value "/~**/*". Other values are not permitted.
header	Yes	string	The HTTP response header which matches the sample.

Example 1

The following example shows how you can create a user-defined HTTP header.

```
{
  "headers": [
    {
      "pattern": "~**/*",
      "header": "Content-Security-Policy: frame-ancestors 'self'
*.somesite.com https://myfriend.site.com;"
    }
  ]
}
```

Example 2

The following example shows how you can reset a user-defined HTTP header.

```
{
  "headers": []
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

The following table shows possible error messages of the WebServer.ChangeResponseHeaders method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read-only	The memory card is write-protected. Therefore, the file cannot be changed.
1301	Invalid pattern	The sample is included in the supported list. Select an approved sample.
1302	HTTP header not allowed	An HTTP header is not contained in the supported list. Select an approved HTTP header.
1303	HTTP header invalid	The HTTP header contains invalid or non-supported characters. Check the HTTP header.
1304	Too many HTTP headers	Too many user-defined HTTP headers. Restrict the number to 1.
1305	Request too large	The entirety of the HTTP header which you want to configure exceeds the maximum permissible size. Reduce the length of the HTTP header.

6.7 Setting the Web server default page

The Web server of the CPU will use the default page if you do not specify a path along with the requested IP address, the domain and the host name in the web browser. You no longer have to configure the default page using TIA Portal. This also eliminates the need to download the hardware configuration when changes are made. You can configure any existing web application as a default page.

6.7.1 WebServer.SetDefaultPage

With this method you set the default page of the web server.

To call the WebServer.SetDefaultPage method, you require "change_webserver_default_page" authorization.

Structure of the request

The following table contains information about the parameters of the request:

Table 6-21 Webserver_SetDefaultPage_Request (object)

Name	Required	Data type	Description
default_page	Yes	string	The default page that you want to set. An empty string undoes the changes and activates the hardware configuration.

Example 1

In the following example, the default page named "index.html" is set for the web application "webapp1".

```
{
  "default_page": "/~webapp1/index.html"
}
```

Example 2

In the following example, the default page is set for the web application "webapp1".

```
{
  "default_page": "/~webapp1/"
}
```

Example 3

In the following example, the default page of the hardware configuration is used.

```
{
  "default_page": ""
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

The following table shows possible error messages of the WebServer.SetDefaultPage method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read-only	The system is currently in a write-protected state. Changes are not currently permitted.
1300	Invalid default page	The default page entered is invalid. Check the entered parameter.

6.7.2 WebServer.ReadDefaultPage

This method is used to read the default page specified with WebServer.SetDefaultPage.

No authorization is required to call the WebServer.ReadDefaultPage method.

Response structure

The following table shows you the structure of server responses to successful requests.

Table 6-22 Webserver_ReadDefaultPage_Response

Name	Required	Data type	Description
default_page	Yes	string	The default page currently configured during runtime. The returned value does not take into account the configuration of the hardware configuration overview in TIA Portal.

6.8 Web applications that can be loaded by the user

With the Web server as of firmware version V2.9, you can use web applications that can be loaded by the user (user-defined). In the following, we will refer to "web applications" for short.

Web applications offer you a set of methods to manage web applications via the Web API. You can use all available Web API methods within a web application.

NOTE

HTTP range requests

For access to web application resources, the web Server provides you with a limited support for HTTP range requests. These requests allow you to read individual areas of a resource.

Advantages

Web applications offer you various advantages compared to the previous method that provided customer pages via the system function SFC 99 in STEP 7. They represent a higher-performance successor to UP:

- The resources are saved in the associated web application. Via the Web API you can download the resources to your PC, edit them and upload them back to the CPU. This procedure results in significantly reduced development times of customer pages.
- Unlike the previous customer pages, you can test the web application during implementation without having to load it onto the CPU.
- You can access resources independent of the CPU mode (e.g. RUN, STOP) and update these.
- Web applications are also available in the STOP mode of the CPU.
- No synchronization between the user program and Web server required by the SFC 99 instruction.
- You can access multiple CPUs within a web application using the Web API.
- The CPU supports saving multiple web applications that you can access simultaneously via the HTTP end point.

- Access to the resources of a web application via the HTTP end point can be activated or deactivated per each application. In so doing, an administrator can deactivate access to a web application, for example, to update the resources consistently.
- You can specify a standard entry page for each application. When you visit the basic URL of a web application, such as https://[ip]/~[application_name], you are automatically being forwarded to the configured home page.
- You can specify an individual media type (MIME type) for each resource.
- You can specify a visibility flag for each resource:
  - Public resource: Access without user authentication
  - Protected resource: Access limited to authenticated users with access right "open_user_pages"
- You can specify for each resource:
  - The time stamp of the resource change
  - The value for the HTTP header ETag

  This enables the caching of resource files by the web browser. This reduces the communication load on the CPU and reduces the loading time of your web application.

NOTE

When you use the Web API for managing web applications, the TIA Portal project in the `SIMATIC.S7S` folder on the SIMATIC load memory changes. Your TIA Portal project is extended by the option of saving resources (e.g. HTML, CSS, JavaScript, etc.) in the project but outside of the data blocks of the user program. Resources are stored securely in the `s7p.web.apps` folder. Include this folder in the backup as well.

As with the previous customer pages, the web applications must not contain any instructions in the STEP 7 user program and are thus purely static files without dynamic content.

Methods for managing web applications

The following methods are available to manage web applications using Web API:

Table 6-23 Methods for managing web applications

Method	Explanation
WebApp.Create (Page 125)	Enables users to create a new web application.
WebApp.Delete (Page 126)	Enables users to delete an existing web application.
WebApp.Rename (Page 127)	Enables users to change the name of an existing web application.
WebApp.Browse (Page 128)	Enables users to display a list of web applications with the associated properties.
WebApp.SetState (Page 131)	Enables users to activate /deactivate a web application for access from the HTTP end point.
WebApp.SetDefaultPage (Page 132)	Enables the user to define a default page for the web application.
WebApp.SetNotFoundPage (Page 133)	Enables users to define a page that is loaded when a requested resource does not exist in the web application.
WebApp.SetNotAuthorizedPage (Page 135)	Enables users to define a page that is loaded when a requested resource is not public (protected) in the web application.

Methods for managing resources

The following methods are available to manage the resources of a web application using Web API:

Table 6-24 Web applications: Methods for managing resources

Method	Explanation
WebApp.BrowseResources (Page 136)	Enables users to display all resources with their properties that are assigned to a web application.
WebApp.CreateResource (Page 138)	Enables users to create a new resource in a web application.
WebApp.DeleteResource (Page 140)	Enables users to delete an existing resource in a web application.
WebApp.RenameResource (Page 141)	Enables users to change the name of an existing resource in a web application.
WebApp.DownloadResource (Page 142)	Allows the user to download a resource from a web application from the CPU.
WebApp.SetResourceVisibility (Page 144)	Enables users to change the visibility of a resource in a web application.
WebApp.SetResourceETag (Page 145)	Enables users to change or delete the ETag value of a resource in a web application.
WebApp.SetResourceMediaType (Page 147)	Enables users to change the media type of a resource in a web application.
WebApp.SetResourceModificationTime (Page 148)	Enables users to set the modification time of a resource in a web application.
WebApp.SetVersion (Page 149)	Enables users to assign a version of the web application that can be loaded as string
WebApp.SetUrlRedirectMode (Page 151)	Enables users to set the redirect mode for default page, non-authorized page and non-found page.

End point for web applications

Web applications are only accessible via secure HTTPS communication. This increases security when accessing the resources of the CPU. When the Web server was configured for HTTP access, requests via HTTP are automatically rerouted to an HTTPS connection.

A web application is accessible via the following URL, in as far as the default page (default_page) or the substitute page of an application (not_found_page) is configured:

```
https://[ip]/~[application_name]
```

A resource of a web application can be reached via the following URL:

```
https://[ip]/~[application_name]/[resource_name]
```

NOTE

The tilde symbol "~" is important in the path and must always exist for web applications.

NOTE

If a web application is disabled, the web server returns the following error code.

```
503 SERVICE UNAVAILABLE
```

Rules for a valid application name

The application name may be max. 100 letters/characters long. The following letters and characters are permitted for the application name:

```
A-Z, a-z, 0-9, -_.+"
```

Rules for a valid resource name

The resource name can be max. 200 letters/characters in length. The following letters and characters are permitted for the resource name:

```
A-Z, a-z, 0-9, -_.+()/|,*!'"
```

Rules for a valid name for the media type

The web browser needs a media type to display a file correctly or to open it. The format of the media type is standardized. For example, "text/html" or "image/jpeg" are valid media types. You can use all valid media types.

More information can be found on the Internet by entering "MIME type" or "media type".

Rules for a valid version number

The version number name can be a max. of 100 letters/characters long. The following letters and characters are permitted for the version number:

```
A-Z, a-z, 0-9, -_.+"
```

Examples:

- V1
- 1.0.0
- 20.0.0.0_0.0.0.0
- WebApp_Test_V1.0

Modes for implementing web applications

Various options are available to implement a web application. There might be special requests for the end point behavior with regard to loading default resources (default page, non-authorized page, non-found page)

If the user, for example, requests a resource which does not exist on the server, the HTTP protocol offers two modes for treating HTTP requests:

- Forward mode: Forwarding of requests
- Redirect mode: Redirecting of requests

Forward mode

If the web server determines that a default treatment is required, the web server searches the target file and immediately returns their content instead of the resource requested originally

by the user. The web server returns the `HTTP-Statuscode 200 OK`, `401 UNAUTHORIZED` or `404 NOT FOUND`.

The web server does not know that another source file was loaded, the original URL remains intact.

If the web browser reloads the resulting page, the original request with the original URL is repeated.

NOTE

This mode is supported as of firmware version V4.0.

Because modern web applications as a rule rewrite the web browser URL, the web server offers the option to treat these requests accordingly as forwarding. An example: The user opens the URL https://s7-1500-webserver/~system/overview in the web browser. No resource with the specified name is available on the server. Therefore the server returns the content of the configured default page (e. g. index.html). The web browser application then takes care of the corresponding processing of the request.

Redirect mode

A redirect is a two-step process. As response to the first request the web server returns the `HTTP-Statuscode 307 TEMPORARY REDIRECT` which also contains the new URL. With this response the web server instructs the web browser to call a second URL which is different from the original URL.

When a web browser loads the second URL again, the original request is not repeated, rather the second URL is requested.

Redirects are slightly slower than forwards because they require two HTTP requests, not just one.

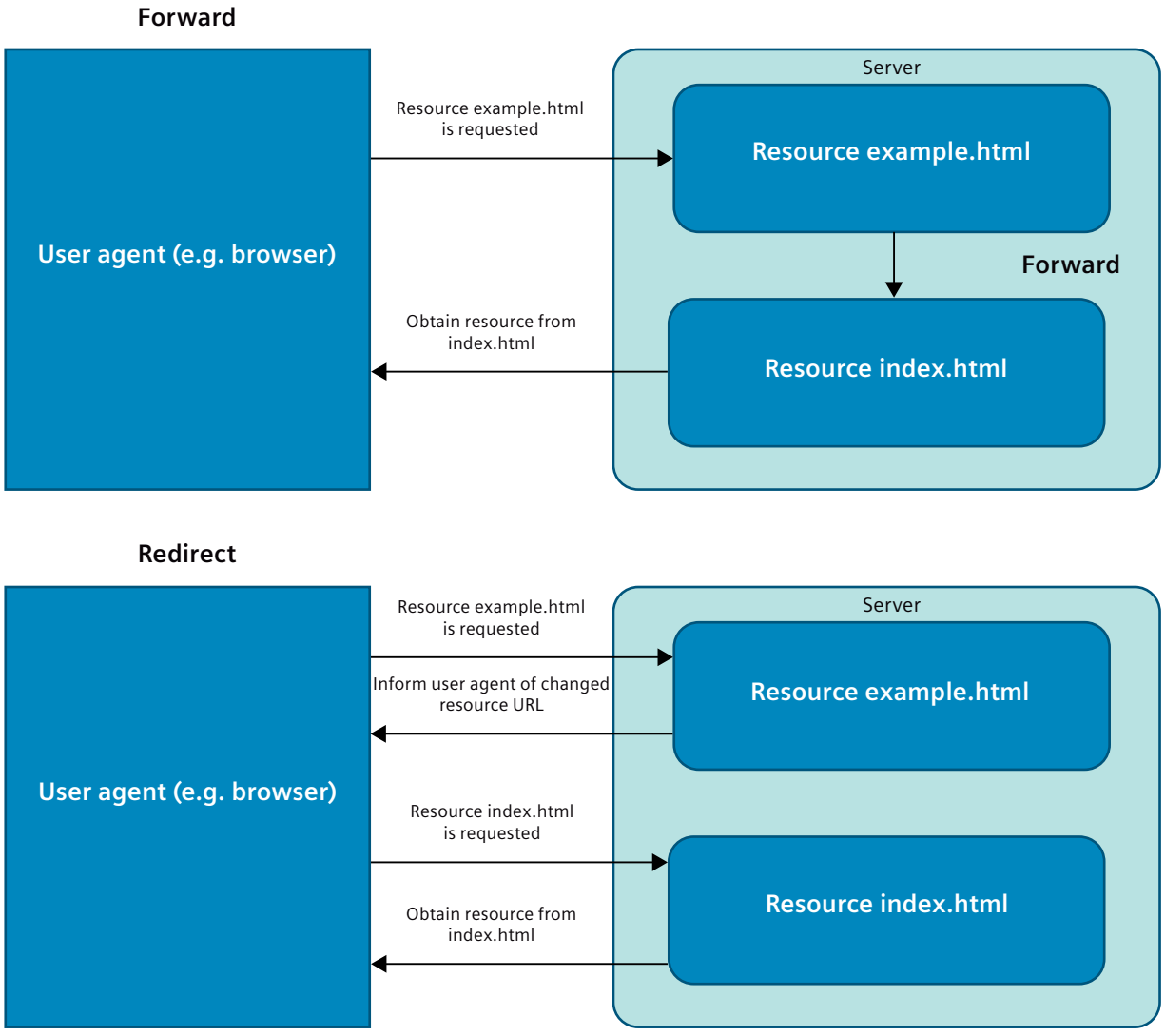


Figure 6-5 Redirect mode

6.8.1 Interaction between web applications

The system web pages, web applications that can be loaded by the user, and VoT (View of Things) applications base on the Web API technology. You can use these web applications with each other without having to authenticate yourself for each application. Other settings, such as language, light/dark user interface mode and time settings, can also be used across the board.

A session cookie is required for the session. The session cookie must be passed to the web server as an X-AUTH token HTTP response header.

Some keys in the local memory are used together with other web applications on the web server. When different web applications are used together, you can either change settings or respond to changes via the keys.

Keys for the interaction between web applications

The following keys are supported for the interaction with other web applications:

Session cookie: Web API token

Name of the key: siemens_web_api_token

The response of the API method Api.Login contains a Web API token. The Web API token is required for each subsequent request to the Web API that depends on an API token. After a log-off the session cookie is removed.

The following example shows the required settings for the parameters SameSite and Secure. This allows you to also use the same API token from other web applications.

```
siemens_web_api_token="E+ty89DBnYhJ6NBxA0a0mkefG2PZ";
SameSite=Strict; Secure
```

Session cookie: Access to web applications

Name of the key: siemens_web_secure

This session cookie allows you to access the contents of the non-public (protected) web application.

If resources of a web application are protected and you want to access data of this web application, then you have to authenticate yourself first, for example on a public page.

For example, perform the following step on the default page:

- From JavaScript, call the Api.Login method using the "include_web_application_cookie" parameter.

Result: If authentication is successful, the Api.Login method returns the session token and a cookie for accessing the protected resources of a web application (see also section Api.Login (Page 85)).

JavaScript is used to set the cookie "web_application_cookie" with a value from the HTTP response of the login as cookie "siemens_web_secure".

NOTE

To ensure that the API token is not lost after the web page is called up again, you can save the API token by means of JavaScript in a cookie or in the local storage of the web browser.

The behavior after timeout for protected web applications corresponds to the behavior after timeout for Web API sessions. After a timeout, the cookie becomes invalid for access to the protected web applications. Reloading a resource file of a web application does not extend the session. Use appropriate methods of the Web API to stay logged in (see section Web API sessions (Page 82)).

The following example shows the required settings for the parameters SameSite and Secure.

```
siemens_web_secure=TXlMdWdnYWdlSGFzVGhlU2FtZSE=; SameSite=Lax;
Secure
```

Local storage: Display of the user interface

Name of the key: siemens_web_ui_theme

With this setting you change the default setting of the web browser. You can ensure the optimal legibility under all light conditions by switching over from the light representation to a dark representation. You can store and use the selected representation of the user interface in the local storage of the web browser.

Table 6-25 Modes for the display of the user interface

Modes for the display of the user interface	Description
auto	The web application uses the display preferred by the web browser.
dark	The web application uses the dark display, irrespective of the configuration of the web browser. This mode uses bright text and bright symbols and graphical operator controls on a dark background.
light	The web application uses the light display, irrespective of the configuration of the web browser. This mode uses dark text and bright symbols and graphical operator controls on a dark background.

Local storage: Language of the user interface

Name of the key: siemens_web_ui_language

If you want to change the language of the user interface, change the language by means of this key. You can store and use the selected language of the user interface in the local storage of the web browser.

Table 6-26 Supported languages of the user interface

Possible settings	Language
de-DE	German
en-US	English (USA)
fr-FR	French
it-IT	Italian
es-ES	Spanish

Possible settings	Language
ja-JP	Japanese
zh-CN	Chinese (Simplified)
ko-KR	Korean
ru-RU	Russian
tr-TR	Turkish
pt-BR	Portuguese (Brazil)

6.8.2 WebApp.Create

With this method, you can create a new web application in the CPU.

To call the WebApp.Create method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-27 WebApp_Create_Request (object)

Name	Required	Data type	Description
name	yes	string	The name of the user-defined web application.
state	no, default is "enabled"	string	The status of the application is: • "disabled": Pages cannot be reached via HTTP end point • "enabled": Pages can be reached via HTTP end point

Example

In the following example, the user creates an application with the name "application_1".

```
{
  "name": "application_1",
  "state": "enabled"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the WebApp.Create method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
500	Application name already exists	An application with the name already exists. Assign a name that does not exist yet.
502	Application limit reached	The maximum number of web applications has been reached. Delete applications that you do not need to free up resources for new applications.
503	Invalid application name	The name of the application is invalid. Assign an application name that meets the rules for a valid application name (see section Web applications that can be loaded by the user (Page 117)).

6.8.3 WebApp.Delete

With this method, you delete an existing web application that can be loaded by the user with all its web page resources.

To call the WebApp.Delete method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-28 WebApp_Delete_Request (object)

Name	Required	Data type	Description
name	yes	string	The name of the web application that is deleted

Example

In the following example, the user deletes an application with the name "application1"

```
{
  "name": "application1"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the WebApp.Delete method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
3	System is busy	The desired operation cannot be performed because the system is currently performing a different request. Restart the query as soon as the current operation is complete.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
501	Application does not exist	An application with this name does not exist.

6.8.4 WebApp.Rename

With this method, you change the name of the web application to a new name.

To call the WebApp.Rename method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-29 WebApp_Rename_Request (object)

Name	Required	Data type	Description
name	yes	string	The current name of the web application.
new_name	yes	string	The new name of the web application.

Example

In the following example, the user changes the name of the application from "application1" to "swac".

```
{
  "name": "application1"
  "new_name": "swac"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the WebApp.Rename method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
500	Application name already exists	An application with the name already exists. Assign a name that does not exist yet.
501	Application does not exist	An application with this name does not exist.
503	Invalid application name	The name of the application is invalid. Assign an application name that meets the rules for a valid application name (see section Web applications that can be loaded by the user (Page 117)).

6.8.5 WebApp.Browse

This method delivers a list of all web applications with the associated properties.

No authorization is required to call the WebApp.Browse method.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-30 WebApp_Browse_Request (object)

Name	Required	Data type	Description
name	No, default is an empty string	string	If this parameter does not exist, all applications will be returned by the method. If the parameter is available, the list will contain only the application whose name matches this parameter. If you have not specified a name, the list may also be empty, depending on whether applications are present or not.

Example

In the following example, the user specifies the name of the "application1" application for the list.

```
{
  "name": "application1"
}
```

Response structure

The following tables show the structure of server responses to successful requests.

Table 6-31 WebApp_Browse_Request (object)

Name	Required	Data type	Description
max_applications	yes	number	Maximum number of applications supported by the CPU
applications	yes	array of `WebApp_Browse_Application_Response`	List of the applications in the CPU

Table 6-32 WebApp_Browse_Application_Response (object)

Name	Required	Data type	Description
name	yes	string	The name of the application
state	yes	string	The status of the application The following values are possible: • "disabled": Pages cannot be reached via HTTP end point • "enabled": Pages can be reached via HTTP end point
type	yes	string	The type of application The following values are possible: • "user": Can be loaded by the user • "vot": View of Things (VoT) web application • "system_builtin": Installed system web sites
version	No	string	The version of the application
redirect_mode	Yes	string	Redirect mode The following values are possible: • "redirect" • "forward" You can find more information on the redirect mode Web applications that can be loaded by the user (Page 117).
default_page	No	string	Default page of the application when no resource name was specified when accessing the web application
not_found_page	No	string	Substitute page in an application when the requested resource does not exist.
not_authorized_page	No	string	If the user has tried to access a protected resource to which the user does not have access. This page can, for example, be implemented as a login page.

Example 1

In the following example, the user requests responses from the server, for no configured version and for an old loaded project.

```
{
  "max_applications": 4,
  "applications": [
    {
      "name": "application1",
      "state": "enabled",
```

```
      "type": "user",
      "default_page": "index.html",
      "redirect_mode": "redirect"
    },
    {
      "name": "vottheapp",
      "state": "enabled",
      "type": "vot",
      "redirect_mode": "redirect"
    }
  ]
}
```

Example 2

In the following example, the user requests responses from the server, for a configured version and for a new project.

```
{
  "max_applications": 4,
  "applications": [
    {
      "name": "application1",
      "state": "enabled",
      "type": "user",
      "version": "V1.23",
      "redirect_mode": "forward",
      "default_page": "index.html"
    },
    {
      "name": "system",
      "state": "enabled",
      "type": "system_builtin",
      "version": "V1.0.0",
      "redirect_mode": "forward"
      "default_page": "index.html",
      "not_found_page": "index.html",
      "not_authorized_page": "index.html"
    }
  ]
}
```

Possible error messages

The following table shows possible error messages of the WebApp.Browse method.

Error code	Error message	Meaning
501	Application does not exist	An application with this name does not exist.

6.8.6 WebApp.SetState

With this method you activate or deactivate a web application that can be loaded or a system web page. A web application or system web page that can be loaded by the user and is deactivated cannot be called by the HTTP end point.

To call the WebApp.SetState method, you require "manage_user_pages" or "manage_system_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-33 WebApp_SetState_Request (object)

Name	Required	Data type	Description
name	yes	string	Name of the web application whose status is changed
state	yes	string	The status of the application is: • disabled; pages cannot be reached via HTTP end point • enabled; pages can be reached via HTTP end point

Example

In the following example, the user deactivates the application with the name "webapp":

```
{
  "name": "webapp",
  "state": "disabled"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the WebApp.SetState method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
501	Application does not exist	An application with this name does not exist.

6.8.7 WebApp.SetDefaultPage

With this method, you set a default page for a user-defined web application. This page is loaded if you have not assigned a resource name when accessing the web application.

To call the WebApp.SetDefaultPage method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-34 WebApp_SetDefaultPage_Request (object)

Name	Required	Data type	Description
name	yes	string	The name of the web application for which the default page is to be configured
resource_name	yes	string	The name of the resource in the web application that can be loaded by the user An empty character string indicates that the default page is to be deleted.

Example

In the following example, the "index.html" page is used as a resource in the "application_1" web application.

```
{
  "name": "application_1",
  "resource_name": "index.html"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the WebApp.SetDefaultPage method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	The requested resource does not exist in the application. When calling this method, select a resource in the application.
505	Resource visibility must be public	The requested resource is not marked as "public". You should change the resource to "public" or select another resource that is already "public".

6.8.8 WebApp.SetNotFoundPage

With this method, you set a page for a web application that can be loaded. This page is loaded if you have used a resource name that does not exist when accessing the web application.

To call the WebApp.SetNotFoundPage method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-35 WebApp_SetNotFoundPage_Request (object)

Name	Required	Data type	Description
name	yes	string	The name of the web application whose page must be changed
resource_name	yes	string	The name of the resource in the web application that can be loaded by the user An empty character string specifies that the PageNotFound page should no longer be set.

Example

In the following example, the "404.html" page is used as a resource in the "application_1" web application.

```
{
  "name": "application_1",
  "resource_name": "404.html"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the WebApp.SetNotFoundPage method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application. Most methods are only allowed for applications of the type "user", however, not for "vot" or "system_builtin".
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	The requested resource does not exist in the application. When calling this method, select a resource in the application.
505	Resource visibility is not public	The requested resource is not marked as "public". You should change the resource to "public" or select another resource that is already "public".

6.8.9 WebApp.SetNotAuthorizedPage

With this method, you set a publicly visible page that can be loaded for a user-defined web application. This page is loaded if you have used a resource name that cannot be accessed publicly (protected) when accessing the web application, thus accessing the web application without a valid cookie.

You can get a valid cookie using the Api.Login (Page 85) method with the "include_web_application_cookie" parameter.

To call the WebApp.SetNotAuthorizedPage method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-36 WebApp_SetNotAuthorizedPage_Request (object)

Name	Required	Data type	Description
name	yes	string	The name of the web application whose public page must be changed
resource_name	yes	string	The name of the resource in the web application that can be loaded by the user An empty character string indicates that the non-public page is to be deleted.

Example

In the following example, the "login.html" page is used as a resource in the "application_1" web application.

```
{
  "name": "application_1",
  "resource_name": "login.html"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the WebApp.SetNotAuthorizedPage method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	The requested resource does not exist in the application. When calling this method, select a resource in the application.
505	Resource visibility must be public	The requested resource is not marked as "public". You should change the resource to "public" or select another resource that is already "public".

6.8.10 WebApp.BrowseResources

This method provides a list of all resources with their properties that are assigned to a web application.

To call the WebApp.BrowseResources method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-37 WebApp_BrowseResources_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the web application whose list is provided
name	No, default is an empty string	string	If this parameter does not exist, all resources must be returned. Otherwise, the list of returned resources only contains one resource whose name matches this parameter. If no such resource is found, then: • The return list must be empty • It is not an error

Example

In the following example, the user specifies the name of the "application1" application for the list.

```
{
  "app_name": "application_1"
}
```

Response structure

The following tables show the structure of the server response to a successful request.

Table 6-38 WebApp_BrowseResources_Response (object)

Name	Required	Data type	Description
max_resources	yes	number	Maximum number of resources supported by the CPU.
resources	yes	array of WebApp_BrowseResources_ Resource_Response	List of the resources in the specific application.

Table 6-39 WebApp_BrowseResources_Resource_Response (object)

Name	Required	Data type	Description
name	yes	string	The name of the resource.
size	yes	integer	The size of the resource in bytes.
media_type	yes	string	The media type of the resource.
etag	No	string	The ETag value of the resource.
visibility	yes	string	The visibility of the resource.
last_modified	yes	string	ISO8601 time stamp as string. The time stamp of the last modification.

Example

In the following example, the user requests responses from the server.

```
{
  "max_resources": 300,
  "resources":
  [
    {
      "name": "index.html",
      "size": 24387,
      "media_type": "text/html",
      "etag": "896a9s8df0897g098a",
      "visibility": "public",
      "last_modified": "2020-08-24T07:08:06Z"
    },
    {
      "name": "secret.js",
      "size": 97826348,
      "media_type": "application/javascript",
      "visibility": "protected",
      "last_modified": "2020-08-24T07:08:06Z"
    }
  ]
```

```
}
```

NOTE

For the CPUs up to firmware version V3.1, the limit of resources is (`max_resources`) 200.

As of firmware version V4.0 the limit is set to 300, in as far as the loaded project is based on V4.0 or higher.

Possible error messages

The following table shows possible error messages of the WebApp.BrowseResources method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	A resource with this name does not exist. When calling this method, select a resource in the application.

6.8.11 WebApp.CreateResource

With this method, you create a new resource in a web application that can be loaded by the user on the CPU.

To call the WebApp.CreateResource method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-40 WebApp_CreateResource_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the web application (Page 117) for which a resource must be created
name	yes	string	The name of the resource (Page 117) that is uploaded
media_type	yes	string	The media type of the resource (Page 117).
visibility	No; the default value is "`public`"	string	The visibility of the resource (Page 144).
etag	no; the default value is an empty character string	string	The ETag value of the resource (Page 145).
last_modified	yes	string	ISO8601 time stamp as string The time stamp of the last modification.

Example

In the following example, the user creates a new resource "index.html" of the application "application_1" with the media type "text/html".

```
{
  "app_name": "application_1",
  "name": "index.html",
  "media_type": "text/html",
  "last_modified": "2020-08-24T07:08:06Z"
}
```

Response structure

This method returns a character string that includes a valid ticket ID. The user uses this ticket to upload the content of the resource to the CPU using the ticket end point.

NOTE

You can create a maximum of one WebApp.CreateResource ticket. You can create a new ticket for this method only after this ticket has been closed.

Example

The following example shows a response for ticket ID.

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

An example of further processing of the ticket ID can be found in the Ticket mechanism (Page 101) section.

Possible error messages

The following table shows possible error messages of the WebApp.CreateResource method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
4	No resources	You have exhausted all tickets in this user session or already have a ticket for WebApp.CreateResource that has not yet been closed. Close existing tickets to free up resources. Then call the method again.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
507	Resource alredy exists	A resource with the specified name already exists for this application. Select a new resource name or delete/rename the resource before you call this method.

Error code	Error message	Meaning
508	Invalid resource name	The resource name is invalid. Correct the resource name (Page 117) before you call this method.
509	Resource limit reached	The maximum number of resources has been exhausted for this application. Delete some resources before you call this method.
511	Invalid modification time	The planned modification time is not within the permissible time window of the modification time. Reduce the modification time accordingly before you call this method.
512	Invalid media type	The media type is invalid. Correct the media type (Page 117) before you call this method.
513	Invalid ETag	The ETag value is invalid. Correct the ETag value (Page 145) before you call this method.

6.8.12 WebApp.DeleteResource

With this method, you can delete a resource from a specific web application.

To call the WebApp.DeleteResource method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-41 WebApp_DeleteResource_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the user-defined web application from which the resource is to be deleted
name	yes	string	The name of the resource that is to be deleted

Example

In the following example, the user deletes the resource "img/wrong.png" of the application "application1".

```
{
  "app_name": "application1",
  "name": "img/wrong.png"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the WebApp.DeleteResource method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
3	System is busy	The desired operation cannot be performed because the system is currently performing a different request. Restart the query as soon as the current operation is complete.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	A resource with the specified name does not exist. When calling this method, select a resource in the application.

6.8.13 WebApp.RenameResource

With this method, you change the name of a resource in a specific web application.

To call the WebApp.RenameResource method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-42 WebApp_RenameResource_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the user-defined web application in which the name of the resource is changed
name	yes	string	The name of the resource that is changed
new_name	yes	string	The new name of the resource.

Example

In the following example, the user changes the name of the resource to "corrspelled.gif" in the "application1" application.

```
{
  "app_name": "application1",
  "name": "msspelled.png",
  "new_name": "corrspelled.gif"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the WebApp.RenameResource method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	A resource with this name does not exist. When calling this method, select a resource in the application.
507	Resource already exists	A resource with the new name already exists for this application. Select a new resource name or delete/rename the resource before you call this method.
508	Invalid resource name	The new resource name is invalid. Correct the resource name (Page 117) before you call this method.

6.8.14 WebApp.DownloadResource

With this method, you can load a resource to a web application that can be loaded by the user from the CPU.

To call the WebApp.DownloadResource method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-43 WebApp_DownloadResource_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the web application that contains the resource
name	yes	string	The name of the resource that is downloaded

Example

In the following example, the user downloads the resource "secrets.pdf" of the application "application_1".

```
{
  "app_name": "application_1",
  "name": "secrets.pdf"
}
```

Response structure

This method returns a character string that includes a valid ticket ID. The user uses this ticket to download the content of the resource from the CPU using the ticket end point.

NOTE

You can create a maximum of one WebApp.DownloadResource ticket at the same time.

Example

The following example shows a response for ticket ID.

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

An example of further processing of the ticket ID can be found in the Ticket mechanism (Page 101) section.

Possible error messages

The following table shows possible error messages of the WebApp.DownloadResource method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
4	No resources	You have exhausted all tickets in this user session or already have a ticket for WebApp.DownloadResource that has not yet been closed. Close existing tickets to free up resources. Then call the method again.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	A resource with this name does not exist. Select a resource name that exists in the application before you call the method.
514	Resource content has been corrupted	The file content has been damaged as a result of file manipulations in the SIMATIC load memory. You can rectify the damage by deleting and recreating the resource file via the API.
504	Resource content is not ready	The content of the resource is not yet ready because it is currently being uploaded. Wait until the upload has finished.

6.8.15 WebApp.SetResourceVisibility

With this method, you change the visibility of a resource, public or protected, in a web application that can be loaded by the user. A protected resource cannot be called by the HTTP end point without authentication.

To call the WebApp.SetResourceVisibility method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-44 WebApp_SetResourceVisibility_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the web application that contains the resource
name	yes	string	The name of the resource that is changed
visibility	yes	string	The visibility of the resource. The following values are possible: public: public protected: only for authorized users An example with a set cookie "web_application_cookie" for access to protected web applications can be found in section Example: Web page for monitoring and controlling a wind turbine (https://support.industry.siemens.com/cs/ww/en/view/107623221/71849212939).

Example

In the following example, the user sets the "secrets.html" resource of the "myapp" application to "protected":

```
{
  "app_name": "myapp",
  "name": "secrets.html",
  "visibility": "protected"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the WebApp.SetResourceVisibility method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	A resource with this name does not exist. Select a resource name that exists in the application before you call the method.
505	Resource visibility must be public	The respective resource is referenced: • as the default page, • as a non-authorized page or • as page not found You cannot mark the resource as protected. You must set the visibility in the application to "public" before calling this method.

6.8.16 WebApp.SetResourceETag

With this method, you change or delete the ETag attribute that is returned when accessing the resource via the HTTP header. ETag (Entity Tag) is an HTTP header field. It only serves to determine changes at the requested resource and is used to avoid redundant data transfers.

To call the WebApp.SetResourceETag method, you require "manage_user_pages" authorization.

Rules for a valid ETag value

You can use any character string as an ETag value as seen in the example below. The length is limited to a maximum of 128 characters.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-45 WebApp_SetResourceETag_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the web application that contains the resource
name	yes	string	The name of the resource that is to be changed.
etag	yes	string	The ETag value of the resource. An empty character string indicates that the value is to be deleted.

Example

In the following example, the user sets the ETag value to "09as7df09h8j23r" for the "secrets.html" resource in the "myapp" application.

```
{
  "app_name": "myapp",
  "name": "secrets.html",
  "etag": "09as7df09h8j23r"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the WebApp.SetResourceETag method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	A resource with this name does not exist. Select a resource name that exists in the application before you call the method.
513	Invalid ETag	The ETag value is invalid. Correct the value before you call this method.

6.8.17 WebApp.SetResourceMediaType

With this method, you change the media type of a resource in a web application that can loaded by the user.

To call the WebApp.SetResourceMediaType method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-46 WebApp_SetResourceMediaType_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the web application that contains the resource
name	yes	string	The name of the resource that is changed
media_type	yes	string	The media type of the resource (Page 117).

Example

In the following example, the user sets the media type to "image/jpeg" for the "secrets.jpg" resource in the "myapp" application.

```
{
  "app_name": "myapp",
  "name": "secrets.jpg",
  "media_type": "image/jpeg"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the WebApp.SetResourceMediaType method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	A resource with this name does not exist. Select a resource name that exists in the application before you call the method.
512	Invalid media type	The media type is invalid. Correct the media type (Page 117) before you call this method.

6.8.18 WebApp.SetResourceModificationTime

With this method, you set the modification time of a resource in a web application that can be loaded by the user.

To call the WebApp.SetResourceModificationTime method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-47 WebApp_SetResourceModificationTime_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the web application that contains the resource
name	yes	string	The name of the resource that is changed
last_modified	yes	string	ISO8601 time stamp as a string; the time stamp of the last change

Example

In the following example, the user sets the modification time to "24.08.2020 07:08:06" for the "secrets.html" resource in the "myapp" application:

```
{
  "app_name": "myapp",
  "name": "secrets.html",
  "last_modified": "2020-08-24T07:08:06Z"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the WebApp.SetResourceModificationTime method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	A resource with this name does not exist. Select a resource name that exists in the application before you call the method.
511	Invalid modification time	The planned modification time is not within the permissible time window of the modification time. Reduce the modification time accordingly before you call this method.

6.8.19 WebApp.SetVersion

With this method you assign a version of the web application that can be loaded as a string

To call the WebApp.SetVersion method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-48 WebApp_SetVersion_Request (object)

Name	Required	Data type	Description
name	yes	string	The name of the web application in which the resource is located.
version	yes	string	The version of the resource in the web application that can be loaded. An empty character string specifies that the version is to be reset. See section Web applications that can be loaded by the user (Page 117) for rules for a valid version number.

Example

In the following example, the user assigns the version "V1.2" to the application with the name "myapp".

```
{
  "name": "myapp",
  "version": "V1.2"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the WebApp.SetVersion method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because this method of the application is not supported either for this application type or for the loaded project version.
501	Application does not exist	An application with this name does not exist.
510	Invalid version string	The version string entered by the user does not fulfill the criteria for valid version strings.

6.8.20 WebApp.SetUrlRedirectMode

With this method, you can set the Redirect mode for default page, non-authorized page and non-found page.

You can find more information on the Redirect mode in the section Web applications that can be loaded by the user (Page 117).

To call the WebApp.SetUrlRedirectMode method, you require "manage_user_pages" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-49 WebApp_SetUrlRedirectMode_Request (object)

Name	Required	Data type	Description
name	Yes	string	The application where you change the Redirect mode.
redirect_mode	Yes	string	Redirect mode of the application.

Example

The following example shows a changed Redirect mode.

```
{
  "name": "myapp",
  "redirect_mode": "redirect"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Example

The following example shows a response of the CPU.

```
true
```

Possible error messages

The following table shows possible error messages of the WebApp.SetUrlRedirectMode method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read-only	The memory card is write-protected. Therefore, the file cannot be changed.
6	Not accepted	The method cannot be executed because this method of the application is not supported either for this application type or for the loaded project version.
501	Application does not exist	An application with this name does not exist.

6.9 Reading and writing process data

6.9.1 Supported data types

Binary representation

The Web API represents the values of simple data types as pure binary data ("raw"). The binary data is formatted as a JSON array. Each element within an array represents a single data byte.

Simple display ("simple")

The Web API formats simple data types into a readable form while retaining the ability to process the data using a program. The following section describes how simple data types are represented as JSON data type.

Supported data types

The following table shows:

- The data types supported by the Web API for reading and writing process values
- The visualization in the Web API
- The respective match of the data type in the TIA Portal

Name of the data type in the TIA Portal	Name of the data type in the Web API	Visualization in Web API
Bool	bool	Boolean JSON value: **true** or **false**
Byte	Byte	JSON unsigned integer in a range from 0 to 255
USInt	usint	
Word	word	JSON unsigned integer in a range from 0 to 65 535

Name of the data type in the TIA Portal	Name of the data type in the Web API	Visualization in Web API
UInt	uint	JSON unsigned integer in a range from 0 to 65 535
HW_ANY	hw_any	
HW_IOSYSTEM	hw_iosystem	
HW_DPMASTER	hw_dpmaster	
HW_DEVICE	hw_device	
HW_DPSLAVE	hw_dpslave	
HW_IO	hw_io	
HW_MODULE	hw_module	
HW_SUBMODULE	hw_submodule	
HW_HSC	hw_hsc	
HW_PWM	hw_pwm	
HW_PTO	hw_pto	
HW_INTERFACE	hw_interface	
HW_IEPORT	hw_ieport	
CONN_ANY	conn_any	
CONN_PRG	conn_prg	
CONN_OUC	conn_ouc	
PORT	port	
RTM	rtm_id	
PIP	pip	
DB_ANY	db_any	
DB_WWW	db_www	
DB_DYN	db_dyn	
DWord	dword	JSON unsigned integer in a range from 0 to 4 294 967 295
UDInt	udint	
AOM_IDENT	aom_ident	
EVENT_ANY	event_any	
EVENT_ATT	event_att	
EVENT_HWINT	event_hwint	
CONN_R_ID	conn_r_id	
LWord	lword	JSON string with a numerical representation of an unsigned integer based on the number 10 in a range from 0 to 18 446 744 073 709 551 615
ULInt	ulint	
SInt	sint	JSON signed integer in a range from -128 to 127
Int	int	JSON signed integer in a range from -32 768 to 32 767
OB_ANY	ob_any	
OB_DELAY	ob_delay	
OB_TOD	ob_tod	
OB_CYCLIC	ob_cyclic	

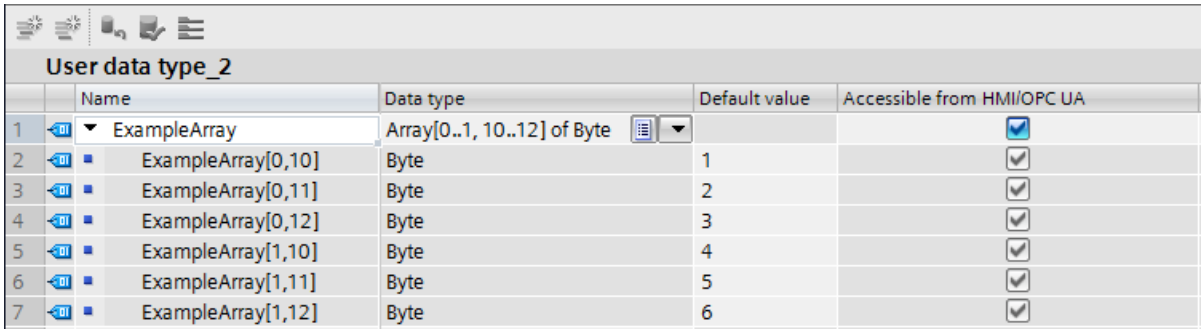
Name of the data type in the TIA Portal	Name of the data type in the Web API	Visualization in Web API
OB_ATT	ob_att	JSON signed integer in a range from -32 768 to 32 767
OB_PCYCLE	ob_pcycle	
OB_HWINT	ob_hwint	
OB_DIAG	ob_diag	
OB_TIMEERROR	ob_timeerror	
OB_STARTUP	ob_startup	
DInt	dint	JSON signed integer in a range from -2 147 483 648 to 2 147 483 647.
LInt	lint	JSON string with a numerical representation of an unsigned integer based on the number 10 in a range from -9 223 372 036 854 775, 808 to 9 223 372 036 854 775 807
Real	real	JSON floating-point number If the floating-point number is infinite or NaN (not-a-number), the Web API returns the value **null** when reading a tag of this type.
LReal	lreal	
Character	char	JSON string with a single ASCII character in a valid range from 0 to 127 If a tag of this type is read with a value outside the valid range, the Web API returns the value **null**.
WChar	wchar	JSON string with a UTF-8 string that represents a single UCS-2 character in a valid range from 0 to 55 295. If a tag of this type is read with a value outside the valid range, the Web API returns the value **null**.
String	string	JSON string with a UTF-8 string If a tag of this type is read with a value outside the valid UTF-8 string (max. length 254 characters), the Web API returns the value **null**.
WString	wstring	JSON string with a UTF-8 string that represents a USC-2 string in a valid range from 0 to 55 295. If a tag of this type is read with a value outside the valid range (max. length 254 characters), the Web API outputs the value **null**.
Date	date	JSON unsigned integer in a range from 0 to 65 535 This value represents the number of days since 01.01.1990.
Time_Of_Day	time_of_day	JSON unsigned integer in a range from 0 to 4 294 967 295 This value represents the number of milliseconds since the beginning of the day.
LTime_Of_Day	ltime_of_day	JSON string with a numerical representation of an unsigned integer based on the number 10 in a range from 0 to 18 446 744 073 709 551 615 This value represents the number of nanoseconds since the beginning of the day.
Time	time	JSON signed integer in a range from -2 147 483 648 to 2 147 483 647) This value represents the number of milliseconds since a user-defined point in time.
LTime	ltime	JSON string with a numerical representation of an unsigned integer based on the number 10 in a range from -9 223 372 036 854 775 808 to 9 223 372 036 854 775 807 This value represents the number of nanoseconds since a user-defined point in time.

Name of the data type in the TIA Portal	Name of the data type in the Web API	Visualization in Web API
S5Time	s5time	JSON object with the keys **basis** and **value**: • The **basis** value is a JSON Unsigned Integer with a value of either 10, 100, 1000, or 10 000. The value **basis** represents the millisecond multiplier of the value **value**. • The value **value** is a JSON unsigned integer in the range from 0 to 999. The value **basis** multiplied by the value **value** gives the timer interval in milliseconds.
Date_And_Time	date_and_time	JSON object with the keys **year**, **month**, **date**, **hour**, **minute**, **second**, and **day_of_week**: • **year** is a JSON unsigned integer with a value in the range from 1 990 to 2 089 • **month** is a JSON unsigned integer with a value in the range 1 to 12 • **day** is a JSON unsigned integer with a value in the range 1 to 31 • **hour** is a JSON unsigned integer with a value in the range 0 to 23 • **minute** is a JSON unsigned integer with a value in the range 0 to 59 • **second** is a JSON floating point number with a value in the range 0 to 60 • **day_of_week** is a JSON string with a value of either **sun**, **mon**, **tue**, **wed**, **thu**, **fri**, or **sat**
LDT	ldt	JSON string with a numerical representation of an unsigned integer based on the number 10 in a range from 0 to 18 446 744 073 709 551 615 This value represents the number of nanoseconds since 01.01.1970 midnight (12:00:00.0 am).
Struct	struct	Structured data type whose data structure can be determined using the PlcProgram.Browse method.
IEC_COUNTER	iec_counter	
IEC_TIMER	iec_timer	
DTL	dtl	
IEC_LTIMER	iec_ltimer	
IEC_SCOUNTER	iec_scounter	
IEC_DCOUNTER	iec_dcounter	
IEC_LCOUNTER	iec_lcounter	
IEC_UCOUNTER	iec_ucounter	
IEC_USCOUNTER	iec_uscounter	
IEC_UDCOUNTER	iec_udcounter	
IEC_ULCOUNTER	iec_ulcounter	
ERROR_STRUCT	error_struct	
NREF	nref	
CREF	cref	

Arrays

Arrays are displayed as JSON objects. The key is a string with a numeric representation of the index.

The following example shows the representation in the TIA Portal:



User data type_2

	Name	Data type	Default value	Accessible from HMI/OPC UA
1	ExampleArray	Array[0..1, 10..12] of Byte		☑
2	ExampleArray[0,10]	Byte	1	☑
3	ExampleArray[0,11]	Byte	2	☑
4	ExampleArray[0,12]	Byte	3	☑
5	ExampleArray[1,10]	Byte	4	☑
6	ExampleArray[1,11]	Byte	5	☑
7	ExampleArray[1,12]	Byte	6	☑

Figure 6-6 Display in the TIA Portal

Addressing the user data

The Web API supports the following addressing formats:

- Symbolic addressing of a tag in the tag table, e.g. Tag_1
- Symbolic addressing of a tag in a data block, e. g. "MyDB".Static_1

6.9.2 Parameter assignment of the block properties

Configuring access to the Web API in the TIA Portal

To restrict the read and write access to data blocks of your project, you can define the desired parameters in the attributes of the respective block.

To allow the Web API to access the data block, activate the check box "DB accessible from Webserver".

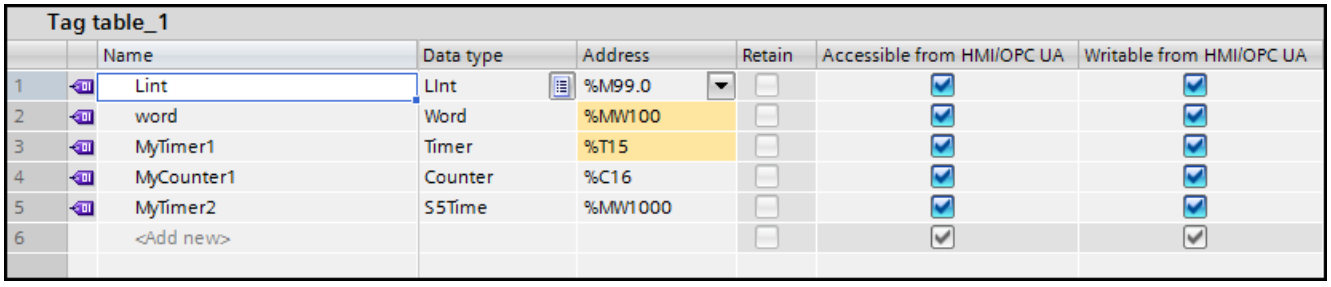
NOTE

F-blocks

Note that fail-safe blocks allow read access only. It is not possible to write tags into fail-safe blocks.

Parameter assignment for access to the Web API in tag tables

For read and write access of Web API to tags, the options "Accessible from HMI/OPC UA/Web API" and "Writable from HMI/OPC UA/Web API" must be activated:



Tag table_1

	Name	Data type	Address	Retain	Accessible from HMI/OPC UA	Writable from HMI/OPC UA
1	Lint	LInt	%M99.0	☐	☑	☑
2	word	Word	%MW100	☐	☑	☑
3	MyTimer1	Timer	%T15	☐	☑	☑
4	MyCounter1	Counter	%C16	☐	☑	☑
5	MyTimer2	S5Time	%MW1000	☐	☑	☑
6	<Add new>			☐	☑	☑

Figure 6-7 Tag table in the TIA Portal

6.9.3 PlcProgram.Read

Use this method to read a single tag from a CPU.

To call the PlcProgram.Read method, you require "read_value" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-50 PlcProgram_Read_Request (object)

Name	Required	Data type	Description
var	yes	string	Name of the tag to be read.
mode	no, default is "simple"	string	Enumeration that determines the response format for this method: • "simple": Returns tag values according to the "simple" representation (see section Supported data types (Page 152)). • "raw": Returns tag values according to the "raw" representation (see section Supported data types (Page 152)).

Example 1

In the following example, the user requests a global tag in the "simple" representation.

```
{
  "var": "\"MotorSpeed\""
}
```

Example 2

In the following example, the user requests a data block tag in the "raw" representation.

```
{
  "var": "\"MyDB\".MyVariable",
  "mode": "raw"
}
```

Response structure

If the request to the server was successful, the server returns JSON data values.

Example 1

The following example shows the result of reading in a tag of type "int" in the "simple" representation.

```
-42
```

Example 2

The following example shows the result of reading in a tag of type "dword" in the "raw" representation.

```
[ 1, 47, 233, 0 ]
```

Possible error messages

The following table shows the possible error messages of the PlcProgram.Read method.

Error code	Error message	Meaning
1	Internal error	An internal error in the desired operation
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
4	No resources	The system does not have the necessary resources to read the requested address. Perform the request again as soon as enough resources are available again.
200	Address does not exist	The requested address does not exist or the Web server cannot access it.
201	Invalid address	The name structure of the symbolic address is not correct.
203	Invalid array index	The dimensions and limits of the array indexes do not correspond to the type information of the CPU.
204	Unsupported address	The data type of the address cannot be read.

NOTE

Tag access with the methods PlcProgram.Read, PlcProgram.Write, and PlcProgram.Browse

With these methods it is not yet possible to access all tags in firmware version V3.0.

There are selective restrictions when reading tags of technology objects. If access to specific tags is not possible, the API returns the message "unsupported".

6.9.4 PlcProgram.Write

This method is used to write a single process tag to the CPU.

To call the PlcProgram.Write method, you require "write_value" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-51 PlcProgram_Write_Request (object)

Name	Required	Data type	Description
var	yes	string	Name of the tag to be written.
value	yes	variant	of the value to be written; The value depends on the operating mode.
mode	no, default is "simple"	string	Enumeration that specifies the format of "value": • "simple": The user must specify the values according to the "simple" representation (see section Supported data types (Page 152)) • "raw": The user must specify the values according to the "raw" representation (see section Supported data types (Page 152))

Example 1

In the following example, the user writes a global tag in the "simple" display.

```
{
  "var": "\"MotorSpeed\"",
  "value": 9001
}
```

Example 2

In the following example, the user writes a tag to a data block in the "raw" representation.

```
{
  "var": "\"MyDB\".MyVariable",
  "value": [ 255, 77, 1, 99 ],
  "mode": "raw"
}
```

Example 3

In the following example, the user writes a string tag consisting of 10 characters to the "simple" representation.

```
{
  "var": "\"MyDB\".MyString",
  "value": "Short Str",
  "mode": "simple"
}
```

Example 4

In the following example, the user writes a string tag consisting of 10 characters with the text "Short Str" in the "raw" representation.

```
{
  "var": "\"MyDB\".MyString",
  "value": [ 10, 9, 83, 104, 111, 114, 116, 32, 83, 116, 114, 0 ],
  "mode": "raw"
}
```

Example 5

In the following example, the user writes a Wstring tag consisting of 6 characters in the "simple" representation.

```
{
  "var": "\"MyDB\".MyWString",
  "value": "Hello",
  "mode": "simple"
}
```

Example 6

In the following example, the user writes a string tag consisting of 6 characters with the text "Hello" in the representation "raw".

```
{
  "var": "\"MyDB\".MyWString",
  "value": [ 0, 6, 0, 5, 0, 72, 0, 101, 0, 108, 0, 108, 0, 111, 0, 0
],
  "mode": "raw"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

The following table shows possible error messages of the PlcProgram.Write method.

Error code	Error message	Meaning
1	Internal error	An internal error in the desired operation
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
4	No resources	The system does not have the necessary resources to write the requested address. Perform the request again as soon as enough resources are available again.
200	Address does not exist	The requested address does not exist or the Web server cannot access the requested address.
201	Invalid address	The name structure of the symbolic address is not correct.
203	Invalid array index	The dimensions and limits of the array indexes do not correspond to the type information of the CPU.
204	Unsupported address	The data type of the address cannot be written.

NOTE

Tag access with the methods PlcProgram.Read, PlcProgram.Write, and PlcProgram.Browse

With these methods it is not yet possible to access all tags in firmware version V3.0.

There are selective restrictions when reading tags of technology objects. If access to specific tags is not possible, the API returns the message "unsupported".

6.9.5 PlcProgram.DownloadProfilingData

S7-1500 CPUs as of firmware version V3.1 support profiling. The PlcProgram.DownloadProfilingData method supplies you with detailed runtime data for the user program in the CPU. The API method returns a ticket that you use to download the runtime data from the CPU. You can evaluate and graphically display the information later in order to analyze the program flow.

This information assists you in the following tasks:

- Runtime optimization of the user program
- Error diagnostics
- Evaluating the power reserve of the automation system
- Quality assurance of the application

To call the PlcProgram.DownloadProfilingData API method, you require the "read_value" authorization.

Response structure

If successful, the method returns a string with a ticket ID. You can use this ticket ID to download the runtime data from the CPU.

For more information about the ticket mechanism, see the Ticket mechanism (Page 101) section.

Example

The following example shows a generated ticket ID for downloading the runtime data.

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

File name

The file name for downloading the runtime data to a web browser is structured as follows:

`[project_name]_[module_name]_YYYY-MM-DD_HH-mm-ss_profiling.bin`

Example: `[1500_example01]_[plc_1]_2023-11-03_12-20-05_profiling.bin`

The file name is returned as an HTTP Content-Disposition header in the server response.

Structure and contents of the file

The file contains a list of entries in which the byte sequence of each entry is based on a specific entry type. The first byte of an entry in each case contains the entry type.

Table 6-52 Block events

Byte	0	1	2	3
0	Entry type	Block type	Block number	
4	Number of OB currently being executed in the program		Call hierarchy of the OB	Priority of the OB
8	Time stamp in 10^{-12} seconds			
12				

The entry types of the following table are supported.

Table 6-53 Entry type

Value	Entry type
0x00	Invalid
0x01	Block event: Start of a block
0x02	Block event: End of a block
0x10	Read process image partition
0x11	Write process image partition
0x20	Direct read access
0x21	Direct write access
0x30	Error ID
0x31	Communication load and cycle time

Entry type: Block events

The file is structured as follows:

The following table explains the meaning of the possible block types:

Table 6-54 Block type

Value	Block type
0x00	Invalid
0x01	Organization block (OB)
0x02	Function (FC)
0x03	Function block (FB)
0x11	System function (SFC)
0x12	System function block (SFB)

The following table provides an example for block events:

Table 6-55 Example for block events

Byte	0	1	2	3
0	0x01	0x02	0x12	0x00
4	0x01	0x00	0x01	0x01
8	0x08	0x07	0x06	0x05
12	0x04	0x03	0x02	0x01

0x01: Start of a block
0x02: Block type = FC
0x0012: Block number = 18
0x0001: OB number = 1
0x01: Call hierarchy of the OB = 1
0x01: Priority of the OB = 1
0x0102030405060708: Time stamp

Entry type: Read/write process image partition

The file is structured as follows:

Table 6-56 Read/write process image partition

Byte	0	1	2	3
0	Entry type	-	Process image partition number	
4	-	-	-	Priority of the OB
8	Time stamp in 10^{-12} seconds			
12				

The following table provides an example for read process image partition:

Table 6-57 Example for read/write process image partition

Byte	0	1	2	3
0	0x10	-	0x01	0x00
4	-	-	-	0x01
8	0x08	0x07	0x06	0x05
12	0x04	0x03	0x02	0x01

0x10 Read process image partition
0x0001: Process image partition number = 1
0x01: Priority of the OB = 1
0x0102030405060708: Time stamp

Entry type: Direct read/write access

The file is structured as follows:

Table 6-58 Direct read/write access

Byte	0	1	2	3
0	Entry type	Event type	DB number	
4	-			Priority of the OB
8	Time stamp in 10^{-12} seconds			
12				

The following table explains the meaning of the possible entry types for direct I/O access:

Table 6-59 Event type

Value	Event type
0x00	Invalid
0x01	I/O
0x02	Motion Control
0x03	Diagnostics

The following table provides an example for direct reading:

Table 6-60 Example for direct reading

Byte	0	1	2	3
0	0x20	0x02	0x12	0x00
4	-	-	-	0x01
8	0x08	0x07	0x06	0x05
12	0x04	0x03	0x02	0x01

0x20: Direct reading
0x02: Motion Control = 2
0x0012: DB number = 18
0x01: Priority of the OB = 1
0x0102030405060708: Time stamp

Entry type: Error ID

The file is structured as follows:

Table 6-61 Entry in the diagnostics buffer

Byte	0	1	2	3
0	Entry type	-	Error type	
4	-	-	-	-
8	Time stamp in 10^{-12} seconds			
12				

The following table provides an example for a diagnostics buffer entry:

Table 6-62 Example for diagnostics buffer entry

Byte	0	1	2	3
0	0x30	-	0x42	0x29
4	-	-	-	-
8	0x08	0x07	0x06	0x05
12	0x04	0x03	0x02	0x01

0x30: Entry in the diagnostics buffer
0x2942: Error type = I/O read access error
0x0102030405060708: Time stamp

Entry type: Communication load and cycle time

The file is structured as follows:

Table 6-63 Entry in the diagnostics buffer

Byte	0	1	2	3
0	Entry type	-	Communication load in %	
4	Last cycle time in µs			
8	Time stamp in 10^{-12} seconds			
12				

The following table provides an example for the communication load:

Table 6-64 Example for communication load

Byte	0	1	2	3
0	0x31	-	0x12	0x00
4	0x03	0x02	0x01	0x00
8	0x08	0x07	0x06	0x05
12	0x04	0x03	0x02	0x01

0x31: Communication load and cycle time
0x0012: Communication load = 18%
0x00010203: Last cycle time = 66051µs
0x0102030405060708: Time stamp

Possible error messages

The following table shows possible error messages of the PlcProgram.DownloadProfilingData method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
4	No resources	You have exhausted all tickets in this user session. Close existing tickets to free up resources. Then call the method again.

Further processing of runtime data

"SIMATIC Controller Profiling (https://support.industry.siemens.com/cs/ww/en/view/109750245)" allows you to analyze and evaluate the runtime behavior of your user program of a CPU as of firmware version V3.1. You can display and analyze all relevant information graphically with Google Chrome.

NOTE

R/H-CPUs

Profiling is available at the R/H-CPUs as of V4.0.

You read out the data from the CPU with "SIMATIC Controller Profiling" and generate a compressed trace file in JSON format for analysis with the Chrome web browser.

You can also find examples of using API methods in practice on Github under the following repository (https://github.com/siemens/simatic-s7-webserver-api).

6.9.6 PlcProgram.Browse

This method allows you to search for tags and the corresponding metadata according to your individual requirements.

To call the PlcProgram.Browse method, you require "read_value" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-65 PlcProgram_Browse_Request (object)

Name	Required	Data type	Description
var	Yes/no, see "Description" column	string	Name of the tag to be searched. If this attribute is present, it cannot be an empty string. • If "mode" = "var", then this attribute is required. The Browse method searches for the provided tag to retrieve the metadata of the tag. • If "mode" = "children", this attribute is optional. The Browse method searches for the tag and returns a list of child tags and metadata.

Name	Required	Data type	Description
mode	Yes	string	Enumeration that determines the behavior of this method: • "var": Displays information about the specified tag. • "children": Outputs information about the immediate descendants (children) of the specified tags.
type	No	array of string	Possible array entries are: • "code_blocks": Reads out all code blocks • "data_blocks": Reads out all data blocks • "tags": Reads out all tags If no "type" parameter is selected for compatibility reasons, only DBs and tags are returned.

Example 1

In the following example, the user searches the root node ("root") of the CPU.

```
{
  "mode": "children"
}
```

Example 2

In the following example, the user searches the descendants (children) of a data block.

```
{
  "var": "\"MyDB\"",
  "mode": "children"
}
```

Example 3

In the following example, the user requests information about a specific tag.

```
{
  "var": "\"MyDB\".MyStruct.MyField",
  "mode": "var"
}
```

Example 4

In the following example, the user searches code blocks of a CPU.

```
{
  "mode": "children",
  "type": ["code_blocks"]
}
```

Example 5

In the following example, the user searches the data blocks of a CPU.

```
{
  "mode": "children",
  "type": ["data_blocks"]
}
```

Example 6

In the following example, the user searches the data blocks, code blocks, and tags of a CPU.

```
{
  "mode": "children",
  "type": ["data_blocks, "code_blocks", "tags"]
}
```

Response structure

The following tables show the structure of server responses to successful requests.

Table 6-66 PlcProgram_Browse_Response (array of objects)

Name	Required	Data type	Description
name	Yes	string	Tag name; can be used as a string identifier for the field.
address	No	string	Address of the tag in STEP 7; only applicable for the tags in the ranges M, I, Q, timer and counter and tags in non-optimized data blocks. The representation corresponds to the addresses in the watch tables in the TIA Portal.
read_only	No	bool	Query whether the tag is read-only. The only valid value is "true". If the tag is to be written, this attribute does not appear.
has_children	No	bool	Query whether the tag is a structured tag with child tags. The only valid value is "true". If the tags are an unstructured data type, this attribute is not displayed.
db_number	No	number	Numerical data block identifier Appears when "datatype" = "datablock" and for each child element of a data block (with corresponding data block number).
area	No	string	Letter which defines the range (M/I/Q/timer/counter) of the tag. If the tag is not in one of these ranges, the attribute does not appear.
datatype	Yes	string	Data type of the tag For data blocks, this is the "datablock" data type; for tags, see the table in the section Supported data types (Page 152). If the data type is not supported, the data type is "unsupported".
max_length	No	number	If the data type is a "string" or "wstring", this value is the maximum number of characters allowed in the tag.

Name	Required	Data type	Description
array_dimensions	No	PlcProgram_Browse_Response_Array-Data	Object arrays arranged from the most significant to the least significant array dimension. The attribute is only displayed if the tag is an array.
block_number	No	number	Number of the logic block
block_type	No	string	The type of the logic block: • "ob" • "fc" • "fb" • "sfc" • "sfb"

Table 6-67 PlcProgram_Browse_Response_ArrayData (array of objects)

Name	Required	Data type	Description
start_index	Yes	integer	Start index for this array dimension, as specified in the TIA Portal project.
count	Yes	integer	Number of elements in this array dimension

Example 1

The following example shows the result of searching the root node "root" of the CPU without code blocks. If the "type" parameter does not exist, only data blocks and tags are output.

```
[
  {
    "name": "TestDB",
    "has_children": true,
    "db_number": 2,
    "datatype": "datablock"
  },
  {
    "name": "GenUsrMsg_Ret",
    "address": "%MW43",
    "area": "M",
    "datatype": "int"
  }
]
```

Example 2

The following example shows the result of searching the descendants (children) of the data block "MyDB" with db_number = 1.

```
[
  {
    "name": "Static_1",
    "db_number": 1,
    "datatype": "int"
  },
  {
    "name": "Static_2",
    "db_number": 1,
    "datatype": "int"
  }
]
```

Example 3

The following example shows the result of querying information via a specific tag.

```
[
  {
    "name": "MyDateTimeValue",
    "db_number": 2,
    "datatype": "date_and_time",
    "array_dimensions": [
      {
        "start_index": 0,
        "count": 3
      }
    ]
  }
]
```

Example 4

The following example shows the result of searching the data blocks at the CPU.

```
[
  {
    "name": "TestDB",
    "block_number": 2,
  },
  {
    "name": "MotorControlConveyorDB",
    "block_number": 23,
  }
]
```

Example 5

The following example shows the result of searching the code blocks at the CPU.

```
[
  {
    "name": "MainOB",
    "block_number": 1,
    "block_type": "ob"
  },
  {
    "name": "MotorControlConveyor",
    "block_number": 23,
    "block_type": "fb"
  }
  {
    "name": "CREATE_DB",
    "block_number": 86,
    "block_type": "sfc",
  }
  {
    ...
  }
]
```

Example 6

The example below shows the result of searching the root node "root" of the CPU with data blocks, code blocks and tags, when the parameter "type" and all 3 possible array entries "data_blocks", "code_blocks" and "tags" are selected.

```
[
  {
    "name": "TestDB",
    "has_children": true,
    "db_number": 2,
    "datatype": "datablock"
  },
  {
    "name": "GenUsrMsg_Ret",
    "address": "%MW43",
    "area": "M",
    "datatype": "int",
  }
  {
    "name": "MainOB",
    "block_number": 1,
    "block_type": "ob"
  }
]
```

Possible error messages

The following table shows possible error messages of the PlcProgram.Browse method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
3	System is busy	The desired operation cannot be performed because the system is currently performing a different request. Restart the query as soon as the current operation is complete.
4	No resources	The system does not have the required resources to retrieve the type information. Perform the request again as soon as enough resources are available again.
200	Address does not exist	The requested address does not exist or the Web server cannot access the requested address.
201	Invalid address	The name structure of the symbolic address is not correct.
202	Variable is not a structure	It is not possible to search the specific address because the tag is not a structured data type.
203	Invalid array index	The dimensions and limits of the array indexes do not correspond to the type information of the CPU.

NOTE

Tag access with the methods PlcProgram.Read, PlcProgram.Write, and PlcProgram.Browse

It is not yet possible to access all tags with these methods in firmware version ≥ V3.0.

There are selective restrictions when reading tags of technology objects. If access to specific tags is not possible, the API returns the message "unsupported".

6.10 Reading and changing the operating mode

6.10.1 Plc.ReadOperatingMode

With this method you read the operating mode of the CPU.

To call the Plc.ReadOperatingMode method, you require "read_diagnostics" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-68 Plc_ReadOperatingMode_Request (object)

Name	Required	Data type	Description
redundancy_id	Yes, for R/H-CPUs; No for all other CPUs	number	The parameter redundancy ID must be available if the request is performed on an R/H-CPU. The redundancy ID has the value 1 or 2. With all other CPUs, the parameter must not be part of the request. You can find more information on the system state SYNCUP at the R/H-CPUs in the System Manual Redundant System S7-1500R/H (https://support.industry.siemens.com/cs/ww/en/view/109754833).

Response structure

If the request to the server was successful, the server returns the operating mode as a string. The following strings are possible for operating modes:

Operating mode	String	Only for R/H-CPU?
STOPSELFINITIALIZATION	stop_self_init	No
STOP	`stop`	No
STARTUP	`startup`	No
RUN	`run`	No
RUNREDUNDANT	run_redundant	Yes

Operating mode	String	Only for R/H-CPU?
HALT	`hold`	No
SYNCUP	syncup	Yes
RUNSYNCUP	run_syncup	Yes
REMOTEUNKNOWN	remote_unknown	Yes
ERRORSEARCH	`unknown`	Yes

Example

```
"stop"
```

Possible error messages

The following table shows possible error messages of the Plc.ReadOperatingMode method.

Error code	Error message	Meaning
-32602	Invalid params	Invalid parameters of the method. This error occurs when the parameter redundancy_id is specified at a non-R/H-CPU or, respectively, when the parameter is invalid at an R/H-CPU.
1	Internal error	An internal error in the desired operation
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.

6.10.2 Plc.RequestChangeOperatingMode

With this method, you request a new operating mode for the CPU.

Note that this is only a request for an operating mode. The conditions for an operating mode change must be given at the CPU, e.g. by the corresponding position of the mode selector. You can use the Plc.ReadOperatingMode (Page 174) method to check whether the operating mode change on the CPU was successful.

To call the Plc.RequestChangeOperatingMode method, you require "change_operating_mode" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-69 Plc_RequestChangeOperatingMode_Request (object)

Name	Required	Data type	Description
mode	yes	string	Requested operating mode: "stop" STOP mode "run" RUN mode
redundancy_id	Yes, for R/H-CPUs; No for all other CPUs	number	The parameter redundancy ID must be available if the request is performed on an R/H-CPU. The redundancy ID has the value 1 or 2. With all other CPUs, the parameter must not be part of the request. You can find more information on the system state SYNCUP at the R/H-CPUs in the System Manual Redundant System S7-1500R/H (https://support.industry.siemens.com/cs/ww/en/view/109754833).

Example 1

In the following example, the RUN operating state is requested at a non-R/H-CPU.

```
{
  "mode": "run"
}
```

Example 2

In the following example, the RUN operating state is requested at an R/H-CPU.

```
{
  "mode": "run",
  "redundancy_id": 1
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

The following table shows possible error messages of the Plc.RequestChangeOperatingMode method.

Error code	Error message	Meaning
-32602	Invalid params	Invalid parameters of the method. This error occurs when the parameter redundancy_id is specified at a non-R/H-CPU or, respectively, when the parameter is invalid at an R/H-CPU.
1	Internal error	An internal error in the desired operation
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.

6.10.3 Plc.ReadModeSelectorState

This method reads the current position of the mode switch at the CPU.

To call the Plc.ReadModeSelectorState method, you require "read_diagnostics" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-70 Plc_ReadModeSelectorState_Request (object)

Name	Required	Data type	Description
redundancy_id	Yes, for R/H-CPUs; No for all other CPUs	number	The parameter redundancy ID must be available if the request is performed on an R/H-CPU. The redundancy ID has the value 1 or 2. With all other CPUs, the parameter must not be part of the request. You can find more information on the system state SYNCUP at the R/H-CPUs in the System Manual Redundant System S7-1500R/H (https://support.industry.siemens.com/cs/ww/en/view/109754833).

Response structure

The following table shows you the structure of server responses to successful requests.

Table 6-71 Plc_ReadModeSelectorState_Response (object)

Name	Required	Data type	Description
mode_selector	Yes	string	Possible values: • "run" • "stop" • "no_switch" • "unknown"

Example

In the following example, the RUN operating mode.

```
{
  "mode_selector": "run"
}
```

Possible error messages

The following table shows possible error messages of the Plc.ReadModeSelectorState method.

Error code	Error message	Meaning
-32602	Invalid params	Invalid parameters of the method. This error occurs when the parameter redundancy_id is specified at a non-R/H-CPU or, respectively, when the parameter is invalid at an R/H-CPU.
1	Internal error	An internal error in the desired operation
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.

6.11 Changing time settings via Web API

6.11.1 Plc.ReadSystemTime

This method returns the system time of the CPU. If you have synchronized the system time of the CPU, for example via the TIA Portal function "Online & diagnostics", the system time corresponds to Coordinated Universal Time (UCT).

No authorization is required to call the Plc.ReadSystemTime method.

Response structure

The following table provides information about the individual parameters of the request.

Table 6-72 Plc_ReadSystemTime_Response (object)

Name	Required	Data type	Description
timestamp	Yes	string	ISO8601 time stamp as string in nanoseconds

Example

The following example shows the structure of the time stamp.

```
{
 "timestamp": "2012-04-23T18:25:43.123456789Z"
}
```

6.11.2 Plc.SetSystemTime

Use this method to set the system time of the CPU (CPU local time).

To call the Plc.SetSystemTime method, you require "change_time_settings" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-73 Plc_SetSystemTime_Request (object)

Name	Required	Data type	Description
timestamp	Yes	string	ISO 8601 timestamp as a string in nanoseconds; represents the time stamp of the system time to be set

Example

The following example shows the structure of the time stamp.

```
{
  "timestamp": "2023-11-05T18:25:43.515154511Z"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

The following table shows possible error messages of the Plc.SetSystemTime method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
900	Invalid timestamp	The time stamp used does not match the required time-stamp format (ISO time-stamp defaults).
901	Time not within allowed time range	The time stamp is not within the allowed period for time stamps. The end of the possible timespan is 2200-12-31T23:59:59.999999999Z

6.11.3 Plc.ReadTimeSettings

This method returns the currently active time, the deviation of the time zone from Coordinated Universal Time (UCT), and any daylight saving time rules.

No authorization is required to call the Plc.ReadTimeSettings method.

Structure of the request

The following tables contain information about the individual parameters of the request.

Table 6-74 Plc_ReadTimeSettings_Response (object)

Name	Required	Data type	Description
current_offset	Yes	string the time range, shown in accordance with ISO 8601	The currently active deviation of the CPU local time from the coordinated universal time (UTC) configured on the CPU in minutes, e.g. "PT1H30M" for a deviation of 1 hour and 30 minutes.
utc_offset	Yes	string the time range, shown in accordance with ISO 8601	The deviation of the time zone from the coordinated universal time (UTC) in minutes without consideration of the daylight saving time rules.
rule	No	object of type Plc_ ReadTimeSettings_ Rule_Response	Displays the daylight saving time settings. If no settings are active, daylight saving time is not set and only the "utc_offset" value is used to calculate the CPU local time.

Table 6-75 Plc_ReadTimeSettings_Rule_ Response (object)

Name	Required	Data type	Description
std	Yes	object of type Plc_ ReadTimeSettings_ StdRule_Response	Stands for the zone time setting.
dst	Yes	object type Plc_ ReadTimeSettings_ DstRule_Response	Stands for the daylight saving time settings

Table 6-76 Plc_ReadTimeSettings_StdRule_Response (object)

Name	Required	Data type	Description
start	Yes	object of type Plc_ ReadTimeSettings_ Start_Response	Stands for the start time of the zone time.

Table 6-77 Plc_ReadTimeSettings_DstRule_Response (object)

Name	Required	Data type	Description
start	Yes	object of type Plc_ ReadTimeSettings_ Start_Response	Stands for the start time of daylight saving time.
offset	Yes	string the time range, shown in accordance with ISO 8601	Stands for the deviation of daylight saving time from the zone time in minutes (only used in the dst object).

Table 6-78 Plc_ReadTimeSettings_Start_Response (object)

Name	Required	Data type	Description
month	Yes	number	Used in the "dst.start/std.start" objects. Stands for the month in which the time starts.
week	Yes	number	Used in the "dst.start/std.start" objects. Stands for the week in which the time change is performed. Value 1: First occurrence of the day of the week in the month Value 5: Last occurrence of the day of the week in the month
day_of_week	Yes	string	Used in the "dst.start/std.start" objects. Describes the day of the week on which the time change is performed as a string with 3 characters. The following values are possible: • sun • mon • tue • wed • thu • fri • sat
hour	Yes	number	Used in the "dst.start/std.start" objects. Describes the hour in which the time change is performed.
minute	Yes	number	Used in the "dst.start/std.start" objects. Describes the minute in which the time change is performed.

Example 1

A daylight saving time rule is configured in the following example.

```
{
 "current_offset": "PT1H",
 "utc_offset": "PT2H",
 "rule":
 {
  "dst":
  {
   "offset": "PT1H",
   "start":
   {
    "month": 4,
    "week": 4,
    "day_of_week": "sun",
    "hour": 3,
    "minute": 0
  }
 },
 "std":
  {
   "start":
```

```
    {
     "month": 10,
     "week": 5,
     "day_of_week": "sun",
     "hour": 2,
     "minute": 0
    }
   }
  }
}
```

Example 2

In the following example, no daylight saving time rule has been configured and a time offset of 1 hour exists.

```
{
 "current_offset": "PT1H",
 "utc_offset": "PT2H",
}
```

Example 3

In the following example, no daylight saving time rule has been configured and a time offset of minus 1 hour and 30 minutes exists.

```
{
  "current_offset": "-PT1H30M",
  "utc_offset": "-PT30M"
}
```

Example 4

In the following example, no daylight saving time rule has been configured and no time offset exists.

```
{
  "current_offset": "PT0M",
  "utc_offset": "PT0M"
}
```

6.11.4 Plc.SetTimeSettings

With this method you can set the time settings of the CPU.

To call the Plc.SetTimeSettings method, you require "change_time_settings" authorization.

Structure of the request

The following tables contain information about the individual parameters of the request.

Table 6-79 Plc_SetTimeSettings_Response (object)

Name	Required	Data type	Description
utc_offset	Yes	string	ISO 8601 timespan as a string; the deviation of the time zone from the coordinated universal time (UTC) without consideration of the daylight saving time rules.
rule	No	object of type Plc_SetTimeSettings_Rule_Response	Displays the daylight saving time settings. If no settings are active, daylight saving time is not set and only the "utc_offset" value is used to calculate the local time.

Table 6-80 Plc_SetTimeSettings_Rule_Response (object)

Name	Required	Data type	Description
std	Yes	object of type Plc_SetTimeSettings_StdRule_Response	Stands for the zone time setting.
dst	Yes	object of type Plc_SetTimeSettings_DstRule_Response	Stands for the daylight saving time settings

Table 6-81 Plc_SetTimeSettings_StdRule_Response (object)

Name	Required	Data type	Description
start	Yes	object of type Plc_SetTimeSettings_Start_Response	Stands for the start time of the zone time.

Table 6-82 Plc_SetTimeSettings_DstRule_Response (object)

Name	Required	Data type	Description
start	Yes	object of type Plc_SetTimeSettings_Start_Response	Stands for the start time of daylight saving time.
offset	Yes	string	ISO 8601 timespan as a string; the deviation of daylight saving time from the zone time in minutes (only used in the dst object).

Table 6-83 Plc_SetTimeSettings_Start_Response (object)

Name	Required	Data type	Description
month	Yes	number	Is used in the "dst.start/std.start" objects. Stands for the month in which the time starts.
week	Yes	number	Is used in the "dst.start/std.start" objects. Stands for the week in which the time change is performed Value 1: First occurrence of the day of the week in the month Value 5: Last occurrence of the day of the week in the month

Name	Required	Data type	Description
day_of_week	Yes	string	Is used in the "dst.start/std.start" objects. Describes the day of the week on which the time change is performed as a string with 3 characters. The following values are possible: • sun • mon • tue • wed • thu • fri • sat
hour	Yes	number	Is used in the "dst.start/std.start" objects. Describes the hour in which the time change is performed.
minute	Yes	number	Is used in the "dst.start /std.start" objects. Describes the minute in which the time change is performed.

Example 1

The following example shows how to set a rule with daylight saving time parameters.

```
{
 "utc_offset": "PT2H",
 "rule":
 {
  "dst":
  {
   "offset": "PT1H",
   "start":
   {
    "month": 4,
    "week": 4,
    "day_of_week": "wed",
    "hour": 3,
    "minute": 0
   }
  },
  "std":
  {
   "start":
   {
    "month": 10,
    "week": 5,
    "day_of_week": "wed",
    "hour": 2,
    "minute": 0
   }
```

```
    }
  }
}
```

Example 2

The following example shows the setting of a rule without daylight saving time parameters.

```
{
 "utc_offset": "PT6H",
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

The following table shows possible error messages of the Plc.SetTimeSettings method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read-only	The system cannot be written to (SIMATIC load memory is write-protected). Changes are currently not permitted.
902	Invalid time rule	The rule provided is invalid. Check that the time settings are correct.
903	Invalid UTC offset	The Coordinated Universal Time (UTC) deviation provided is invalid. Check that the time settings are correct.

6.12 Reading diagnostics and service data

6.12.1 Project.ReadLanguages

This method returns a list with the project languages available on the CPU.

You can then use the "Alarms.Browse" or "DiagnosticBuffer.Browse" methods in one of the available languages to get alarm messages or diagnostic messages in the available languages.

To call the Project.ReadLanguages method, you require "read_diagnostics" authorization.

Available languages

The available languages are interpreted with the language codes to RFC4647. The following languages can be available for the CPUs:

Language code to RFC4647	Language
de	German (Germany)
en	English (US)
fr	French (France)
es	Spanish (Spain)
it	Italian (Italy)
ja	Japanese (Japan)
zh	Chinese (China)
ko	Korean (Korea)
ru	Russian (Russia)
tr	Turkish (Türkiye)
pt	Portuguese (Portugal)

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-84 Project_ReadLanguages_Request (object)

Name	Required	Data type	Description
mode	No	string	This parameter specifies the languages that can be returned: • "all" (all languages are listed) • "active" (only active languages are listed)

Response structure

The following tables show the structure of server responses to successful requests.

Table 6-85 Project_ReadLanguages_Response (Array of objects)

Name	Required	Data type	Description
languages	Yes	array of Project_ReadLanguages_Entry_Response	Object array, where each object represents a project language.

Table 6-86 Project_ReadLanguages_Entry_Response (object)

Name	Required	Data type	Description
language	Yes	string	String with the project language as a language code to RFC4647
active	Yes	bool	This parameter specifies whether the project language is available for API methods, for example Alarms.Browse (Page 189).
user_interface_languages	Yes	array of string	String array, with each string representing an interface language for which the project language is to be used as the default. An application can overrule this information.

Example 1

The following example shows the output of a project language as a string for a firmware version <= 3.1.

```
{
  "languages":
  [
    {
      "language": "en-US"
    }
  ]
}
```

NOTE

No project language configured

If no project language was configured, the array is empty.

Example 2

The example below shows several languages with different status for a firmware version >= 4.0.

```
{
  "languages":
  [
    {
      "language": "en-US",
      "active": true,
      "user_interface_languages": [
        "de-DE",
        "en-US"
      ]
    },
    {
      "language": "en-US",
      "active": false,
      "user_interface_languages": []
    }
  ]
}
```

Possible error messages

The following table shows possible error messages of the Project.ReadLanguages method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.

6.12.2 Alarms.Browse

With this method you can determine which alarms are currently active on the CPU, and when the last change occurred within the diagnostics buffer.

To call the Alarms.Browse method, you require "read_diagnostics" authorization.

Structure of the request

The following tables contain information about the individual parameters of the request.

Table 6-87 Alarms_Browse_Request (object)

Name	Required	Data type	Description
language	Yes	string	The desired language in which the text is returned in RFC 4647 format, for example "en-US"
count	No	number	The maximum number of alarm entries that are returned. The default value is 50. If you want to determine the current status of the diagnostics buffer, enter 0 as "count".
alarm_id	No	string	The alarm ID of the CPU for which you are requesting data. If the Alarm ID is included, only the "count" parameter can be offered as a filter.
filters	No	object of type Alarms_Browse_Filters_Request	Optional object containing parameters for filtering the response

Table 6-88 Alarms_Browse_Filters_Request (object)

Name	Required	Data type	Description
mode	Yes	string	The mode that determines whether attributes are to be included or excluded in the response. The following modes are available: • include • exclude
attributes	Yes	array of strings	Possible array entries are: • "alarm_text" • "info_text" • "status" • "timestamp" • "acknowledgement" • "alarm_number" • "producer"

Example 1

The following example shows a request for reading a single alarm with all alarm areas in the English language:

```
{
  "language": "en-US",
  "alarm_id": "1231231231"
}
```

Example 2

The following example shows the request for reading a single alarm without the alarm areas excluded under "exclude":

```
{
  "language": "en-US",
  "alarm_id": "1231231231",
  "filters":
  {
    "mode": "exclude",
    "attributes": ["alarm_text", "info_text"],
  }
}
```

Example 3

The following example shows the request for reading 50 alarms with the alarm ranges included in "include".

```
{
  "language": "en-US",
  "count": 50,
  "filters":
  {
    "mode": "include",
    "attributes": ["status", "acknowledgement"]
  }
}
```

Response structure

The following tables show the structure of server responses to successful requests.

Table 6-89 Alarms_Browse_Response (object)

Name	Required	Data type	Description
language	No, optional and only available when entries are returned; not relevant if, for example, count = 0 is requested	string	The language of the response in which the message text is output. If no valid language was requested, the server outputs the message "invalid".
last_modified	Yes	string	ISO 8601 time stamp in UTC as string; The time stamp with the last change of the alarm system since you made the last read request. The time stamp allows you to see when the last alarm change occurred in the system without having to check individual alarms in detail.

Name	Required	Data type	Description
count_current	Yes	number	The number of active alarms.
count_max	Yes	number	The maximum number of active alarms.
entries	No	array of objects of Alarm_Browse_Entry_response	The list of pending alarms, where each alarm entry is represented as an object.

Table 6-90 Alarms_Browse_Entry_Response (object)

Name	Required	Data type	Description
id	Yes	string	Alarm ID.
alarm_number	Yes, if not excluded by user	number	Alarm number.
status	Yes, if not excluded by user	string	Alarm status; is either "incoming" or "outgoing".
timestamp	Yes, if not excluded by user	string	ISO 8601 time stamp as a string; Time stamp in Coordinated Universal Time (UTC) of the time at which the alarm assumed the status "incoming" or "outgoing".
producer	Yes, if not excluded by user	string	Possible alarm trigger: • "program_alarm" • "system_diagnostics" • "motion" • "security" • "sinumerik" • "graph7" • "prodiag" • "other"
hwid	No	number	Contains the hardware ID if the alarm producer is "system_diagnostics".
acknowledgement	No	object	Appears when an alarm needs to be acknowledged.
acknowledgement.state	Yes	string	String in readable form, which provides information about the status of the acknowledgement: • "not_acknowledged" • "acknowledged"
acknowledgement.timestamp	Yes	string	ISO 8601 time stamp as a string; If the current status ("incoming" or "outgoing") has been acknowledged, the time stamp provides information about the time of acknowledgement.
alarm_text	Yes, if not excluded by user	string	Alarm text in the selected language.
info_text	Yes, if not excluded by user	string	Info text in the selected language.
text_inconsistent	No	boolean	If the alarm text or info text is inconsistent, this flag returns "true".

Example 1

The following example shows the response to a query with a number = 0:

```
{
  "last_modified": "2012-04-23T18:25:43.511546151Z",
  "count_current": 1,
  "count_max": 5000
}
```

Example 2

The following example shows the response to a request with a non-acknowledged alarm and with all attributes (no filters set).

```
{
  "entries":
  [
    {
      "id": "121651651651",
      "timestamp": "2012-04-23T18:25:43.511987654Z",
      "status": "incoming",
      "alarm_number": 37,
      "producer": "system_diagnostics",
      "hwid": 49,
      "acknowledgement":
      {
        "state": "not_acknowledged"
      },
      "alarm_text": "CPU maintenance demanded: Emergency IP suite
parameter for IE interface activated PLC_1516 / Current CPU
operating mode: STOP",
      "info_text": "Short name: CPU general Order number: 6ES7
516-3AP03-0AB0",
      "text_inconsistent": false
    }
  ],
  "last_modified": "2012-04-23T18:25:43.511546151Z",
  "count_current": 1,
  "count_max": 5000,
  "language": "en-US"
}
```

Example 3

The following example shows the response to a request with an acknowledged alarm and with all attributes (no filters set):


```
{
  "entries":
  [
    {
      "id": "121651651651",
      "timestamp": "2012-04-23T18:25:43.511980000Z",
      "status": "incoming",
      "alarm_number": 35,
      "producer": "system_diagnostics",
      "hwid": 49,
      "acknowledgement":
      {
        "state": "acknowledged",
        "timestamp": "2012-04-23T18:25:50.123456789Z"
      },
      "alarm_text": "CPU status message: CPU not in RUN Current CPU
operating mode: STOP",
      "info_text": "Short name: CPU general Order number: 6ES7
516-3AP03-0AB0",
      "text_inconsistent": false
    }
  ],
  "last_modified": "2012-04-23T18:25:43.511546151Z",
  "count_current": 1,
  "count_max": 5000,
  "language": "en-US"
}
```


Example 4

The following example shows the response to a request with a non-acknowledged alarm and with all attributes (no filters set) and with a valid language.


```
{
  "entries":
  [
    {
      "id": "121651651651",
      "timestamp": "2012-04-23T18:25:43.511987654Z",
      "status": "incoming",
      "alarm_number": 37,
      "producer": "program_alarm",
      "alarm_text": "My alarm text created by Program_Alarm",
```

```
      "info_text": "My info text created by Program_Alarm",
      "text_inconsistent": false
    }
  ],
  "last_modified": "2012-04-23T18:25:43.511546151Z",
  "count_current": 1,
  "count_max": 5000,
  "language": "en-US"
}
```


Example 5

The following example shows the response to a request with a non-acknowledged alarm and with all attributes (no filters set) and with an invalid language.


```
{
  "entries":
  [
    {
      "id": "121651651651",
      "timestamp": "2012-04-23T18:25:43.511987654Z",
      "status": "incoming",
      "alarm_number": 37,
      "producer": "program_alarm",
      "alarm_text": "#273, 1",
      "info_text": "#273, 0",
      "text_inconsistent": true
    }
  ],
  "last_modified": "2012-04-23T18:25:43.511546151Z",
  "count_current": 1,
  "count_max": 5000,
  "language": "invalid"
}
```


Example 6

The following example shows the filtered response to a query with only alarm text and info text, partially empty (empty text configured in project).


```
{
  "entries":
  [
    {
      "id": "121651651651",
      "alarm_text": "my alarm text",
      "info_text": ""
```

```
    },
    {
      "id": "121651651652",
      "alarm_text": "",
      "info_text": "my info text"
    }
  ],
  "last_modified": "2012-04-23T18:25:43.511546151Z",
  "count_current": 2,
  "count_max": 5000,
  "language": "en-US"
}
```


Example 7

The following example shows the filtered response to a query with only alarm IDs:


```
{
  "entries":
  [
    {
      "id": "121651651651",
      "producer": "system_diagnostics"
    },
    {
      "id": "121651651652",
      "producer": "system_diagnostics"
    },
    {
      "id": "121651651653",
      "producer": "program_alarm"
    }
  ],
  "last_modified": "2012-04-23T18:25:43.511546151Z",
  "count_current": 3,
  "count_max": 5000,
  "language": "en-US"
}
```

Possible error messages

The following table shows possible error messages of the Alarms.Browse method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
800	Invalid alarm ID	The alarm ID provided is invalid.
801	Invalid parameters	The request is invalid because provided parameters are invalid (e.g. the parameters "count" and "id" are present at the same time).

6.12.3 Alarms.Acknowledge

Use this method to acknowledge individual alarms.

To call the Alarms.Acknowledge method, you require "acknowledge_alarms" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 6-91 Alarms_Acknowledge_Request (object)

Name	Required	Data type	Description
id	Yes	string	The acknowledgement ID of the alarm to be acknowledged. The acknowledgement ID can be found in the alarm object returned by the Alarm.Read method.

Example

The following example shows the ID of an alarm to be acknowledged:

```
{
 "id": "1213515461"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

The following table shows possible error messages of the Alarms.Acknowledge method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.

6.12.4 DiagnosticBuffer.Browse

With this method you read out entries from the diagnostics buffer of the CPU.

To call the DiagnosticBuffer.Browse method, you require "read_diagnostics" authorization.

Structure of the request

The following tables provide information about the required parameters for the request.

Table 6-92 DiagnosticBuffer_Browse_Request (object)

Name	Required	Data type	Description
language	Yes	string	The desired language in which the text is returned in RFC 4647 format, for example "en-US"
count	No	number	The maximum number of alarm entries that are returned. The default value is 50. If you want to determine the current status of the diagnostics buffer, enter 0 as "count".
filters	No	object of type DiagnosticBuffer_ Browse_Filters_ Request	The object that represents the different filtering options.

Table 6-93 DiagnosticBuffer_Browse_Filters_Request (object)

Name	Required	Data type	Description
attributes	Yes	array of strings	The following attributes are possible for the diagnostics buffer entries: • short_text • long_text • help_text
mode	Yes	string	The mode that determines whether attributes are to be included or excluded in the request. The following modes are available: • include • exclude

Example

The following example shows a request of the diagnostic entries as LCID value 1033 (dec value), which stands for "English – United States".

```
{
  "language": "en-US",
  "count": 50,
  "filters":
  {
    "mode": "include",
    "attributes": ["short_text", "long_text", "help_text"]
  }
}
```

Response structure

The following tables show the structure of server responses to successful requests.

Table 6-94 DiagnosticBuffer_Browse_Response (object)

Name	Required	Data type	Description
entries	No	array of objects of DiagnosticBuffer_Browse_Entry_Response	Array of diagnostics buffer entries, where each object represents one diagnostics buffer entry.
last_modified	Yes	string	ISO 8601 time stamp as a string; time stamp of the last change in the diagnostics buffer.
count_current	Yes	number	Number of available diagnostics buffer entries.
count_max	Yes	number	Maximum number of possible diagnostics buffer entries
language	No, optional and only available when entries are returned; not relevant if, for example, count = 0 is requested	string	The language in which the response is output.

Table 6-95 DiagnosticBuffer_Browse_Entry_Response (object)

Name	Required	Data type	Description
timestamp	Yes	string	ISO 8601 time stamp as a string; The attribute is provided in Coordinated Universal Time (UTC) and not in CPU local time.
status	Yes	string	The status parameter for events is either "incoming" (incoming events) or "outgoing" (outgoing events).
event	Yes	object	Contains the event ID of the diagnostics buffer which consists of the text list ID and the text ID of the event
long_text	Yes, if not excluded by the user	string	Diagnostics buffer entries in the long form
short_text	Yes, if not excluded by the user	string	Diagnostics buffer entries in the short form
help_text	Yes, if not excluded by the user	string	Help text for an incoming event.

Table 6-96 DiagnosticBuffer_Browse_Entry_Event_Response (object)

Name	Required	Data type	Description
event.textlist_id	Yes	number	Text list ID of the event
event.text_id	Yes	number	Text ID of the event

Example 1

The following example shows the representation of an entry for display = 0.

```
{
  "last_modified": "2012-04-23T18:25:43.514678521Z",
  "count_current": 1234,
  "count_max": 3200
}
```

Example 2

The following example shows the representation of a multiline entry in the diagnostics buffer (system diagnostics message).

```
{
  "entries":
  [
    {
      "timestamp": "2012-04-23T18:25:43.511854547Z",
      "status": "outgoing",
      "long_text": "CPU info: Boot up

memory card type: Program card (external load memory)
CPU changes from OFF to STOP (initialization) mode

PLC_2 / PLC_2",
      "short_text": "Boot up - CPU changes from OFF to STOP
(initialization) mode",
      "help_text": "",
      "event":
      {
        "textlist_id": 2,
        "text_id": 16385
      }
    }
  ],
  "last_modified": "2012-04-23T18:25:43.514678521Z",
  "count_current": 1234,
  "count_max": 3200,
  "language": "en-US"
}
```

Example 3

The following example shows the representation of a multiline entry in the diagnostics buffer (system diagnostics message).

```
{
  "entries":
  [
    {
      "timestamp": "2012-04-23T18:25:43.511854547Z",
      "status": "incoming",
      "long_text": "My long text created with Gen_UsrMsg!",
      "short_text": "My short text created with Gen_UsrMsg!",
      "help_text": "",
      "event":
      {
        "textlist_id": 512,
        "text_id": 42
      }
    }
  ],
  "last_modified": "2012-04-23T18:25:43.514678521Z",
  "count_current": 1234,
  "count_max": 3200,
  "language": "en-US"
}
```

Example 4

The following example shows the representation for a valid language but with an invalid text list ID / text ID.

```
{
  "entries":
  [
    {
      "timestamp": "2012-04-23T18:25:43.511854547Z",
      "status": "incoming",
      "long_text": "#254, 128",
      "short_text": "#253, 128",
      "help_text": "",
      "event":
      {
        "textlist_id": 0,
        "text_id": 42
      }
    }
  ],
```

```
  "last_modified": "2012-04-23T18:25:43.514678521Z",
  "count_current": 1234,
  "count_max": 3200,
  "language": "en-US"
}
```

Example 5

The following example shows the display for a valid language but with an invalid text list ID / text ID in the text.

```
{
  "entries":
  [
    {
      "timestamp": "2012-04-23T18:25:43.511854547Z",
      "status": "incoming",
      "long_text": "The status of the sensor is: #513, 0",
      "short_text": "The status of the sensor is: #513, 0",
      "help_text": "",
      "event":
      {
        "textlist_id": 512,
        "text_id": 1
      }
    }
  ],
  "last_modified": "2012-04-23T18:25:43.514678521Z",
  "count_current": 1234,
  "count_max": 3200,
  "language": "en-US"
}
```

Example 6

The following example shows the representation for an invalid language.

```
{
  "entries":
  [
    {
      "timestamp": "2012-04-23T18:25:43.511854547Z",
      "status": "incoming",
      "long_text": "#254, 17",
      "short_text": "#253, 17",
      "help_text": "#32770, 4416",
      "event":
```

```
        {
          "textlist_id": 2,
          "text_id": 4416
        }
      }
    ],
    "last_modified": "2012-04-23T18:25:43.514678521Z",
    "count_current": 1234,
    "count_max": 3200,
    "language": "invalid"
}
```

Possible error messages

The following table shows possible error messages of the DiagnosticBuffer.Browse method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.

6.12.5 Modules.DownloadServiceData

This method returns a ticket that you use to download service data from the CPU. You can then forward the service data to Customer Support, for example for an analysis of your production data in the event of an error.

To call the Modules.DownloadServiceData method, you require "download_service_data" authorization.

You can find more information about the ticket mechanism in the section Ticket mechanism (Page 101).

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-97 Modules_DownloadServiceData_Request (object)

Name	Required	Data type	Description
hwid	Yes	number	Hardware ID of the module whose service data you want to read out.

Example

The following example shows a request of the method.

```
{
  "hwid": 49
}
```

Response structure

If successful, the method returns a string with a ticket ID. Use this ticket ID to download the service data.

NOTE

You can create a maximum of one Modules.DownloadServiceData ticket for all users. You can create a new ticket for this method only after this ticket has been closed.

Example

The following example shows a generated ticket ID for service data download.

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

Possible error messages

The following table shows possible error messages of the Modules.DownloadServiceData method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
4	No resources	You have exhausted all tickets in this user session. Close existing tickets to free up resources. Then call the method again.
7	Partner not accessible	The data of the CPU of an R/H system is not available. This can be the case when the system is in SYNCUP operating state or RUN-Redundant system state or when the service data of the partner CPU is queried.
600	No service data resources	Only one ticket resource for service data is available for all users at the same time.
1100	Invalid hardware identifier	The specified hardware ID is not valid for the current request. Make sure that you have used the correct hardware ID.

6.13 Motion Control

6.13.1 Technology.BrowseObjects

This method returns a list of all technology objects that were configured for the CPU. You can use this method, for example, to create a diagnostics overview for Motion Control.

To call the Technology.BrowseObjects method, you require "read_diagnostics" authorization.

Response structure

The following tables show the structure of server responses to successful requests.

Table 6-98 Technology_BrowseObjects_Response (object)

Name	Required	Data type	Description
objects	Yes	Array of Technology_ BrowseObjects_ Entry_Response	The array contains a list of all technology objects whereby each element is represented as an object with detail information. If no technology object is configured, an empty list is returned.

Table 6-99 Technology_BrowseObjects_Response (array of objects)

Name	Required	Data type	Description
number	Yes	number	The number of the technology object
name	Yes	string	The name of the technology object
type	Yes	string	The type of the technology object The following values are possible: • "to_speedaxis" • "to_positioningaxis" • "to_synchronousaxis" • "to_externalencoder" • "to_measuringinput" • "to_outputcam" • "to_camtrack" • "to_cam" • "to_kinematics" • "to_leadingaxisproxy" • "to_cam_10k" • "to_interpreter" • "to_interpretermapping" • "to_interpreterprogram" • "to_cam_600seg" • "to_cam_6kseg" • "unknown"
version	Yes	floating-point number	The version number of the technology object

Example

The following example shows the output of the properties of a configured technology object.

```
{
  "objects":
  [
    {
      "number": 2,
      "name": "Kinematics_1",
      "type": "to_kinematics",
      "version": 6
    }
  ]
}
```

Possible error messages

The following table shows possible error messages of the Technology.BrowseObjects method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.

6.13.2 Technology.Read

This method returns the value of a process tag of the technology object. You use this method for Motion Control diagnostics.

To call the Technology.Read method, you require "read_diagnostics" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-100 Technology_Read_Request (object)

Name	Required	Data type	Description
var	Yes	string	Name of the tag to be read. The name must not be empty.
mode	No Default is "simple"	string	Enumeration that specifies the response format of this method: • "simple": Returns tag values according to the "simple" representation • "raw": Returns tag values according to the "raw" representation You can find a description of the values in section Supported data types (Page 152).

NOTE

Format of the request

The format of the request and the response is the same as for the PlcProgram.Read (Page 157) method, but with the constraint that you can only read technology object tags.

Example

In the following example, the user requests a data block tag in the "raw" representation.

```
{
  "var": "\"MyTO_Axis\".StatusWord",
  "mode": "raw"
}
```

Response structure

If the request to the server was successful, the server returns JSON data values.

Example 1

The following example shows the result of reading in a tag of type "int" in the "simple" representation.

```
-42
```

Example 2

The following example shows the result of reading in a tag of type "dword" in the "raw" representation.

```
[ 1, 47, 233, 0 ]
```

Possible error messages

The following table shows possible error messages of the Technology.Read method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
4	No resources	The system does not have the necessary resources to read the requested address. Perform the request again as soon as enough resources are available again.
200	Address does not exist	The requested address does not exist or the Web server cannot access it.
201	Invalid address	The name structure of the symbolic address is not correct.
203	Invalid array index	The dimensions and limits of the array indexes do not correspond to the type information of the CPU.
204	Unsupported address	The data type of the address cannot be read.
1400	Not a technology object	The tag is not a technology object tag. Therefore, the tag cannot be read.

6.14 Backing up and restoring the configuration

6.14.1 Plc.CreateBackup

With this method, you request a ticket to create a backup file of the CPU configuration.

To call the Plc.CreateBackup method, you require "backup_plc" authorization.

Response structure

The method returns a string with a ticket ID for creating a backup file.

Example

The following example shows a generated ticket ID for creating a backup file.

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

An example of further processing of the ticket ID can be found in the Ticket mechanism (Page 101) section.

Possible error messages

The following table shows possible error messages of the Plc.CreateBackup method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
4	No resources	You have exhausted all tickets in this user session. Close existing tickets to free up resources. Then call the method again.
1000	Backup creation in progress	The creation of a backup file is in progress.
1001	Backup restoration in progress	The restoration of a saved configuration is currently being carried out. It is not possible to perform both operations at the same time.
1004	PLC not in STOP	A backup file can only be created when the CPU is in STOP mode. Set the CPU to STOP mode and execute the request again.

Overview of creating a backup file

Perform the steps below to create a backup file:

1. Authenticate yourself with the API method Api.Login.
2. Request a ticket for creating a backup file using the Plc.CreateBackup method.

   If you are authorized to call this method and the CPU is in STOP, the CPU creates a ticket. Once a ticket has been created for the creation of a backup file, it is no longer possible to switch to the RUN. This ensures the consistency of the backup file.
3. Use the ticket end point to start downloading the backup file.

   The CPU informs you about the current status of the generation in the additional ticket information. Additional information is available using the Api.BrowseTickets(id) method.
4. After the download has been completed successfully, the CPU sets the ticket to the "completed" status.
5. Close the ticket using the Api.CloseTicket(id) method.
6. Now you can set the CPU back to RUN

Format of the backup file name

The default file name of the backup file, which is returned by the HTTP content disposition header, contains the following information:

- Time stamp in Coordinated Universal Time (UTC)
- Module name
- Name of the TIA Portal project
- F-collective signature (for F-CPUs)

6.14.2 Plc.RestoreBackup

Use this method to request a ticket that restores the configuration of a CPU via a backup file.

To call the Plc.RestoreBackup method, you require "restore_plc" authorization.

Structure of the request

The following table contains information about the parameters of the request:

Table 6-101 Plc_RestoreBackup_Request (object)

Name	Required	Data type	Description
password	Yes	string	The required password for the logged on user An empty password string is transferred for the "Everybody" or "Anonymous" user. The "password" attribute is therefore always required. The password must be the password of the user who previously authenticated via Api.Login and whose session token was used to call the Plc.RestoreBackup method.

Example

The following example shows how to enter a password.

```
{
 "password": "SecurePassword"
}
```

Response structure

The method returns a string. The string contains a ticket ID that you can use to restore the configuration on a backup file.

Example

The following example shows a generated ticket ID for restoring the configuration of a CPU.

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

An example of further processing of the ticket ID can be found in the Ticket mechanism (Page 101) section.

Possible error messages

The following table shows possible error messages of the Plc.RestoreBackup method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method.
4	No resources	You have exhausted all tickets in this user session. Close existing tickets to free up resources. Then call the method again.
5	System is read-only	The system cannot be written to (SIMATIC load memory is write-protected). Changes are currently not permitted.
1000	Backup creation in progress	The creation of a backup file is in progress.
1001	Backup restoration in progress	The restoration of a saved configuration is currently being carried out. It is not possible to perform both operations at the same time.
1003	Restore not possible through this interface	Calling the method via CM/CP modules, via IP the address or via a virtual CP is not allowed. Perform the recovery via the IP address of one of the network interfaces of a CPU.
1004	PLC not in STOP	A backup file can only be created when the CPU is in STOP mode. Set the CPU to STOP mode and execute the request again.
1005	Legitimation failed	The user legitimation was not successful. Reasons for this can be the entry of password for the "Everybody" or "Anonymous" user, or the entry of an invalid password.

Overview of the recovery of the CPU configuration

The following section shows you all steps that are required to restore the CPU configuration.

NOTE

Tracking the recovery process via Api.BrowseTickets

The Api.BrowseTickets (Page 106) method provides information about the current status of the recovery.

Use this method to found out the recovery process phase, and whether the recovery process was successful.

1. Authenticate yourself with the API method Api.Login.
2. Request a ticket for restoring a CPU configuration using the Plc.RestoreBackup method.

3. Use the ticket end point to start the upload of the backup file.

   The CPU receives the file header and checks whether it is valid. After a successful check, the CPU restarts after 3 seconds. If the file header is invalid, the CPU aborts the restoration process and the ticket changes to the `failed` status.

   When checking the restoration process, the following states may occur. These states can be read out via the API method Api.BrowseTickets.

Status	Description
`waiting`	The waiting state is active until the upload of the backup file is started.
`ongoing`	As soon as the upload of the backup file has started, the state changes to `ongoing`.
`rebooting_format`	The CPU is restarting. The reason for the restart is the formatting of the SIMATIC load memory.
`rebooting`	The CPU is restarting. The reason for the restart is the activation of the restored project.
`failed`	An error occurred during the upload of the backup file. You can abort the upload when an error occurs.
`failed_failsafe`	An error occurred during the execution of the fail-safe function. Ensure that the password passed to the Plc.RestoreBackup method is correct and the user has the F-Admin function right.
`failed_wrong_interface`	You have started the upload of the backup file to the CPU via a CM or CP interface.

NOTE

Loss of the configuration during the restoration process

Note that the CPU loses the configuration during the restoration process.

4. Before restarting the CPU, it is possible to read the status with the Api.BrowseTickets(id) method. The additional information informs you about restarting the CPU and formatting the SIMATIC load memory as next steps.

   To be informed about the process and all messages, we recommend that you read the information of the Api.BrowseTickets method cyclically, e.g. every second.

5. The CPU then restarts and formats the SIMATIC load memory.

   After the restart you can use the Api.Ping method to determine when the CPU is available again.

   During the restart, the Api.BrowseTickets and Api.Ping methods do not respond.

6. The CPU puts the web server into a state with reduced functionality. Only a limited number of API methods is available to you during this time.

   NOTE

   If you want to restore the CPU to its normal state during the restoration process, perform a download via the TIA Portal. After the download, the CPU and the web server are again in normal operation and all functions are available for use.

7. Use the Api.Login method to log on with the logon data that were also valid at the beginning of the restoration process.

   NOTE

   In this state of the CPU only a local authentication is possible. If the restoration was started with a central user (parameter mode was specified with "umc"), this user is available locally in this mode. Always use the restore mode mode=local or leave the parameter out completely.

8. Request a ticket for the restore using the Plc.RestoreBackup method.
9. Upload the backup file via the ticket end point.

   After successful upload, the CPU restarts after 3 seconds.

10. Before restarting the CPU, it is possible to read the status with the Api.BrowseTickets(id) method. The additional information informs you about the restart and about the successful upload.

    The restart is required to activate the new project.

11. During the restart you can use the Api.Ping method to determine when the CPU is available again.
12. As soon as the CPU is available again, the restoration process is completed and the recovered project is loaded into the CPU.

You can now log in with the credentials of a user of the project loaded into the CPU, if desired.

Further work in the API requires a new login with the Api.Login method.

6.15 Accessing contents of the SIMATIC load memory

The methods described in this section allow you to access the files in the file system in the SIMATIC load memory. You can access standard files as well as your own user files (UserFiles), DataLogs, and recipes.

NOTE

Access to the file system for R/H-CPUs

The file API only offers limited access to the SIMATIC load memory. Access is restricted to the "UserFiles", "DataLogs", and "Recipes" folders. Other content in the SIMATIC load memory is not accessible via the API.

6.15.1 Files.Browse

This method returns a list of the contents of subfolders and attributes of a specific folder or a file which are located on the SIMATIC load memory of a CPU.

To call the Files.Browse method, you require "read_file" authorization.

NOTE

For R/H-CPUs, only a maximum of 3 folders – Recipes, UserFiles and DataLogs – are returned in the root folder of the SIMATIC load memory. No other folders are returned.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-102 Files_Browse_Request (object)

Name	Required	Data type	Description
resource	No	string	Path to the folder or file from the root node. For the root node, the use of a "/" is necessary. You can optionally use a "/" for the root node. If the attribute "resource" is missing or empty, the system interprets it as "/".

Example 1

The following example shows a request specifying the desired path to a txt file:

```
{
  "resource": "/myfolder/file.txt"
}
```

Example 2

The following example shows a request specifying the desired path to a csv file (DataLog).

```
{
  "resource": "/Datalogs/datalog1.csv"
}
```

Response structure

The following tables show the structure of server responses to successful requests.

Table 6-103 Files_Browse_Response (object)

Name	Required	Data type	Description
resources	yes	array of files_Browse_Entry_Response	Resource list.

Table 6-104 Files_Browse_Entry_Response (object)

Name	Required	Data type	Description
name	yes	string	Name of the entry.
type	yes	string	Type of entry, either "file" or "dir".
size	No	number	Size of the file in bytes (if type is "file").
last_modified	yes	string	ISO8601 time stamp as string; time stamp of the last change.
state	No	string	Attribute reserved for active or inactive DataLogs in the "DataLogs" folder

Example 1

The following example shows the response to a request with a file and a folder:

```
{
  "resources":
  [
    {
      "name": "my_dir",
      "type": "dir",
      "last_modified": "2012-04-23T18:25:43Z"
    },
    {
      "name": "my_file.txt",
      "type": "file",
      "size": 87654567,
      "last_modified": "2012-04-23T18:25:43Z"
    }
  ]
}
```

Example 2

The following example shows the response to a query to the test.txt file. Note that the entry is case-sensitive.

```
{
  "resources":
  [
    {
      "name": "Test.txt",
      "type": "file",
      "size": 87654567,
      "last_modified": "2012-04-23T18:25:43Z"
    }
  ]
}
```

Example 3

The following example shows the response to a query with an active csv file:

```
{
  "resources":
  [
    {
      "name": "datalog1.csv",
      "type": "file",
      "size": 87654567,
      "last_modified": "2012-04-23T18:25:43Z",
      "state": "active"
    }
  ]
}
```

Possible error messages

The following table shows possible error messages of the Files.Browse method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method.
4	No resources	The system does not have the required resources to carry out this request.
300	Path contains an illegal sequence	The parameter specified under "resource" violates the naming convention (e.g. contains invalid characters).
301	Entity access is restricted	The parameter specified under "resource" is subject to access restrictions.

Error code	Error message	Meaning
302	Entity does not exist	The file or path to be accessed by the "resource" parameter does not exist.
307	Maximum path depth exceeded	The parameter specified under "resource" exceeds the maximum path length.
310	Entity access denied	The parameter specified under "resource" was rejected.

6.15.2 Files.Download

This method creates a ticket that you can use to download a file from the CPU.

To call the Files.Download method, you require "read_file" authorization.

For more information about the ticket mechanism, see the Ticket mechanism (Page 101) section.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-105 Files_Download_Request (object)

Name	Required	Data type	Description
resource	yes	string	Path to the file from the root node of the SIMATIC load memory

Example 1

The following example shows a request specifying the desired path to a txt file:

```
{
  "resource": "/myfolder/file.txt"
}
```

Example 2

The following example shows a request specifying the desired path to a csv file (DataLog).

```
{
  "resource": "/Datalogs/datalog1.csv"
}
```

Response structure

If successful, the method returns a string with a ticket ID.

Example

The following example shows a generated ticket ID for downloading the file:

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

NOTE

Ticket-based file downloads

For all ticket-based file downloads, the ticket returns a file name in the HTTP content disposition header. You can use this file name as the default file name, or as the name used by the web browser as the default name.

Possible error messages

The following table shows possible error messages of the Files.Download method.

Error code	Error message	Meaning
2	Permission denied	The user does not have read permissions.
4	No resources	The system does not have the required resources to carry out this request.
300	Path contains an illegal sequence	The parameter specified under "resource" violates the naming convention (e.g. contains invalid characters).
301	Entity access is restricted	The parameter specified under "resource" is subject to access restrictions.
302	Entity does not exist	The file or path to be accessed by the "resource" parameter does not exist.
303	Entity in use	The file or path to be accessed by the "resource" parameter is locked because of another operation (e.g. because of a write access).
306	Entity not a file	The file name specified under "resource" is attempting to access a folder.
307	Maximum path depth exceeded	The parameter specified under "resource" exceeds the maximum path length.
310	Entity access denied	The parameter specified under "resource" was rejected.

6.15.3 Files.Create

This method creates a ticket that you use to upload a file to the CPU.

To call the Files.Create method, you require "write_file" authorization.

For more information about the ticket mechanism, see the Ticket mechanism (Page 101) section.

NOTE

Uploading a file on R/H-CPUs

If a file is uploaded during SYNCUP, the R/H system aborts the upload of this file. During SYNCUP, the R/H does not wait until the file has uploaded.

NOTE

Storage location of files on R/H-CPUs

Files can only be created in the Recipes and UserFiles folders and not in the root folder or outside these two folders.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-106 Files_Create_Request (object)

Name	Required	Data type	Description
resource	yes	string	Path to the file from the root node of the SIMATIC load memory

Example

The following example shows a request specifying the path to the desired file:

```
{
  "resource": "/mydir/file.txt"
}
```

Response structure

If successful, the method returns a string with a ticket ID.

Example

The following example shows a generated ticket ID for uploading the file:

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

Possible error messages

The following table shows possible error messages of the Files.Create method.

Error code	Error message	Meaning
2	Permission denied	The user does not have read permissions.
4	No resources	The system does not have the required resources to carry out this request.
5	System is read-only	The system cannot be written to at present. File changes are not possible because the SIMATIC load memory is read-only.
300	Path contains an illegal sequence	The parameter specified under "resource" violates the naming convention (e.g. contains invalid characters).
301	Entity access is restricted	The parameter specified under "resource" is subject to access restrictions.
302	Entity does not exist	The folder or subfolder to be accessed by the "resource" parameter does not exist.
304	Entity already exists	The parameter specified under "resource" is attempting to create a file that already exists.
305	Entity not a directory	The folder or subfolder specified under "resource" is attempting to access a file.
307	Maximum path depth exceeded	The parameter specified under "resource" exceeds the maximum path length.
310	Entity access denied	The parameter specified under "resource" was rejected.

6.15.4 Files.Rename

This method changes the name of a file or folder. You can also use this method to move files from one folder to another folder.

To call the Files.Rename method, you require "write_file" authorization.

NOTE

Note that you cannot use this method for the "DataLogs" folder.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-107 Files_Rename_Request (object)

Name	Required	Data type	Description
resource	yes	string	Current file path or folder path
new_resource	yes	string	New file path or folder path

Example

The following example shows a change of the file name.

```
{
  "resource": "/folder/old_file_name.txt",
  "new_resource": "/folder/new_file_name.txt"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

The following table shows possible error messages of the Files.Rename method.

Error code	Error message	Meaning
2	Permission denied	The user does not have write permission for the file.
4	No resources	The system does not have the required resources to carry out this request.
5	System is read-only	The system cannot be written to (SIMATIC load memory is write-protected). Changes are currently not permitted.
300	Path contains an illegal sequence	The parameter specified under "resource" and/or under "new_resource" violates the naming convention (e.g. contains invalid characters).
301	Entity access is restricted	The parameter specified under "resource" and/or under "new_resource" is subject to access restrictions.
302	Entity does not exist	The file or path to be accessed by the parameter "resource" and/or under "new_resource" does not exist.
303	Entity in use	The parameter specified under "resource" is accessing a file or folder that is already locked by another operation (e.g. read or write).
304	Entity already exists	The parameter specified in "new_resource" is attempting to create a file or folder that already exists.
307	Maximum path depth exceeded	The parameter specified under "resource" exceeds the maximum path length.
308	Directories cannot be moved	The parameter specified under "resource" and/or "new_resource" is attempting to move a folder. Moving a folder structure is not allowed.
310	Entity access denied	The parameter specified under "resource" was rejected.

6.15.5 Files.Delete

This method deletes files from the CPU.

To call the Files.Delete method, you require "write_file" authorization.

NOTE

Deleting DataLog files

You can also delete DataLog files with this method, but only if the file is not currently in use. If the DataLog file is currently in use, error message 303 appears: Entity in use.

NOTE

Deleting inactive DataLog files

If you have created a ticket for DataLogs.DownloadAndClear or Files.Download on an inactive DataLog file, you can still use the Files.Delete method to delete this file.

As a result, a download that has already been started or will be started in the future will fail with these tickets.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-108 Files_Delete_Request (object)

Name	Required	Data type	Description
resource	Yes	string	Path to the file from the root node of the SIMATIC load memory

Example 1

The following example shows a request specifying the desired path to a txt file:

```
{
  "resource": "/myfolder/file.txt"
}
```

Example 2

The following example shows a request specifying the desired path to a csv file (DataLog).

```
{
  "resource": "/Datalogs/datalog1.csv"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

The following table shows possible error messages of the Files.Delete method.

Error code	Error message	Meaning
2	Permission denied	The user does not have write permission for the file.
4	No resources	The system does not have the required resources to carry out this request.
5	System is read-only	The system cannot be written to at present. File changes are not possible because the SIMATIC load memory is read-only.
300	Path contains an illegal sequence	The parameter specified under "resource" violates the naming convention (e.g. contains invalid characters).
301	Entity access is restricted	The parameter specified under "resource" is subject to access restrictions.
302	Entity does not exist	The folder or subfolder to be accessed by the "resource" parameter does not exist.
303	Entity in use	The parameter specified under "resource" is accessing a file or folder that is already locked by another operation (e.g. read or write).
306	Entity not a file	The file name specified under "resource" is attempting to access a folder.
307	Maximum path depth exceeded	The parameter specified under "resource" exceeds the maximum path length.
310	Entity access denied	The parameter specified under "resource" was rejected.

6.15.6 Files.CreateDirectory

This method creates a new folder.

To call the Files.CreateDirectory method, you require "write_file" authorization.

NOTE

Available folders for R/H-CPUs

You can only create the DataLogs, Recipes and UserFiles folders. The folders are created on the system with the corresponding spelling UserFiles, Recipes, DataLogs, regardless of whether you specify `resource="/datalogs"`, for example.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-109 Files_CreateDirectory_Request (object)

Name	Required	Data type	Description
resource	yes	string	Path to the file from the root node of the SIMATIC load memory

Example

The following example shows a request specifying the path to the desired folder.

```
{
  "resource": "/SPH_Storage/OPCUA"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

The following table shows possible error messages of the Files.CreateDirectory method.

Error code	Error message	Meaning
2	Permission denied	The user does not have write permissions.
4	No resources	The system does not have the required resources to carry out this request.
5	System is read-only	The system cannot be written to at present. Changes are not possible because the SIMATIC load memory is write-protected.
300	Path contains an illegal sequence	The parameter specified under "resource" violates the naming convention (e.g. contains invalid characters).
301	Entity access is restricted	The parameter specified under "resource" is subject to access restrictions.
302	Entity does not exist	The folder to be accessed by the "resource" parameter does not exist.
304	Entity already exists	The parameter specified under "resource" is attempting to create a folder that already exists.
305	Entity not a direct-ory	The parameter specified under "resource" is attempting to access a file.
307	Maximum path depth exceeded	The parameter specified under "resource" exceeds the maximum path length.
310	Entity access denied	The parameter specified under "resource" was rejected.

6.15.7 Files.DeleteDirectory

This method deletes an existing folder from the CPU.

To call the Files.DeleteDirectory method, you require "write_file" authorization.

NOTE

Recursive deletion

Note that recursive deletion is not possible, and that the folder must be empty before you can delete it.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-110 Files_DeleteDirectory_Request (object)

Name	Required	Data type	Description
resource	yes	string	Path of the folder to be deleted

Example

The following example shows a request specifying the path to the desired folder.

```
{
  "resource": "/SPH_Storage"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

The following table shows possible error messages of the Files.DeleteDirectory method.

Error code	Error message	Meaning
2	Permission denied	The user does not have write permissions.
4	No resources	The system does not have the required resources to carry out this request.
5	System is read-only	The system cannot be written to at present. Changes are not possible because the SIMATIC load memory is write-protected.
300	Path contains an illegal sequence	The parameter specified under "resource" violates the naming convention (e.g. contains invalid characters).
301	Entity access is restricted	The parameter specified under "resource" is subject to access restrictions.
302	Entity does not exist	The folder to be accessed by the "resource" parameter does not exist.
303	Entity in use	The parameter specified under "resource" is accessing a folder that is already locked by another operation (e.g. read or write).

Error code	Error message	Meaning
305	Entity not a directory	The parameter specified under "resource" is attempting to access a file.
307	Maximum path depth exceeded	The parameter specified under "resource" exceeds the maximum path length.
310	Entity access denied	The parameter specified under "resource" was rejected.

6.15.8 DataLogs.DownloadAndClear

This method creates a ticket to download DataLogs from the CPU and delete them after processing.

NOTE

If you do not want to delete the contents of DataLogs after downloading, use the Files.Download method instead.

For more information about this method, see the Files.Download (Page 216) section.

NOTE

You can apply the DataLogs.DownloadAndClear method only if the DataLog is not currently in use. If the DataLog is currently in use, the error message 303: Entity in use appears.

To call the DataLogs.DownloadAndClear method, you require "write_file" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-111 DataLogs_DownloadAndClear_Request (object)

Name	Required	Data type	Description
resource	yes	string	The name of the DataLog you want to download. Alternatively, you can use a path starting with /datalogs/ (see examples below).

Example

The following example shows a request specifying the desired path to a csv file (DataLog).

```
{
  "resource": "/DataLogs/datalog1.csv"
}
```

Response structure

If successful, the method returns a string with a ticket ID.

Example

The following example shows a generated ticket ID for downloading and deleting the DataLog:

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

Possible error messages

The following table shows possible error messages of the DataLogs.DownloadAndClear method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
4	No resources	The system does not have the required resources to carry out this request.
5	System is read-only	The system cannot be written to at present. Changes to the log file are not possible because the SIMATIC load memory is write-protected.
303	Entity in use	The file or path to be accessed by the "resource" parameter is locked because of another operation (e.g. the user program is currently accessing the DataLog).
309	Entity is not a valid data log	The parameter specified under "resource" is attempting to access an unlinked DataLog. An unlinked DataLog cannot be deleted.

6.16 Reading information from SIMATIC Safety

6.16.1 Failsafe.ReadRuntimeGroups

This method outputs a list with all available F-runtime groups.

To call the Failsafe.ReadRuntimeGroups method, you require "read_diagnostics" authorization.

Response structure

The following tables show the structure of server responses to successful requests.

Table 6-112 Failsafe_ReadRuntimeGroups_Response (object)

Name	Required	Data type	Description
groups	Yes	array of Failsafe_ReadRuntimeGroups_Entry_Response	Object array in which each object represents one F-runtime group.

Table 6-113 Failsafe_ReadRuntimeGroups_Entry_Response (object)

Name	Required	Data type	Description
name	Yes	string	Name of F-runtime group
signature	Yes	string	Signature of F-runtime group as array of decimal numbers Each number represents one byte of the signature.
cycle_time_current	Yes	string	ISO 8601 time span as string Current cycle time in milliseconds.

Name	Required	Data type	Description
cycle_time_max	Yes	string	ISO 8601 time span as string Maximum cycle time in milliseconds.
runtime_current	Yes	string	ISO 8601 time span as string Current runtime in milliseconds.
runtime_max	Yes	string	ISO 8601 time span as string Maximum runtime in milliseconds.

Example

The following example shows the parameters of the response to a query with an F-runtime group with remaining time "remaining_time".

```
{
  "groups":
  [
    {
      "name": "RTG_1",
      "signature": "FD62F235",
      "cycle_time_current": "PT0.110S",
      "cycle_time_max": "PT0.200S",
      "runtime_current": "PT0.050S",
      "runtime_max": "PT0.080S"
    }
  ]
}
```

Possible error messages

The following table shows possible error messages of the Failsafe.ReadRuntimeGroups method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.

6.16.2 Failsafe.ReadParameters

With this method, you can read out fail-safe parameters of a fail-safe CPU or a fail-safe module via the hardware ID of the module.

To call the Failsafe.ReadParameters method, you require "read_diagnostics" authorization.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-114 Failsafe_ReadModuleParameters_Request (object)

Name	Required	Data type	Description
hwid	Yes	number	Hardware ID of the module whose parameters you want to read out.

Response structure

The following tables show the structure of server responses to successful requests.

Table 6-115 Failsafe_ReadParameters_Response (object)

Name	Required	Data type	Description
safety_mode	No	string	Status message indicating whether safety mode is active ("enabled") or not ("disabled"). Note that this status message only applies to the CPU and not to other modules.
type	Yes	string	Defines whether the required hardware ID is the fail-safe module or represents a different fail-safe module.
parameters	No	object	Indicates whether the required hardware ID is the fail-safe module with safety program or represents a different fail-safe module. The returned object is alternatively of type: • Failsafe_ReadParameters_Cpu_Response • Failsafe_ReadParameters_Module_Response

Table 6-116 Failsafe_ReadParameters_Cpu_Response (object)

Name	Required	Data type	Description
last_f_program_modification	Yes	string	ISO 8601 time stamp in UTC as string; time stamp of the last change in the safety program.
collective_signature	Yes	string	Collective F signature as byte array with 4 numbers for representing a 32-bit signature.
remaining_time	No	string	ISO 8601 time span as string; Remaining time in milliseconds (as of firmware version V2.9).

Table 6-117 Failsafe_ReadParameters_Module_Response (object)

Name	Required	Data type	Description
f_monitoring_time	Yes	string	ISO 8601 time stamp in UTC as string; F-monitoring time in milliseconds
f_source_address	Yes	number	F-source address
f_destination_address	Yes	number	F-destination address
f_par_crc	Yes	string	CRC signature of the F parameters as a byte array with 4 numbers for representing a 32-bit signature.

Example 1

The following example shows the parameters of a fail-safe CPU without safety mode.

```
{
  "type": "f_cpu"
}
```

Example 2

The following example shows the parameters of a fail-safe CPU in active safety mode.

```
{
  "type": "f_cpu",
  "safety_mode": "enabled",
  "parameters":
  {
    "last_f_program_modification": "2012-04-23T18:25:43.510Z",
    "collective_signature": "C572BC16",
    "remaining_time": "PT5D2H33M5.123S"
  }
}
```

Example 3

The following example shows the parameters of a fail-safe module in active safety mode.

```
{
  "type": "f_module",
  "parameters":
  {
    "f_monitoring_time": "PT0.123S",
    "f_source_address": 123,
    "f_destination_address": 123,
    "f_par_crc": "F062F235"
  }
}
```

Possible error messages

The following table shows possible error messages of the Failsafe.ReadParameters method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
1100	Invalid hardware identifier	The specified hardware ID is not valid for the current request. Make sure that you have used the correct hardware ID.

6.17 Reading CPU information

6.17.1 Plc.ReadCpuType

With this method, you can read the product names and the article number of the CPU. Thus you have the possibility to find out before the authentication with which CPU you are communicating.

No authorization is required to call the Plc.ReadCpuType method.

Response structure

The following table shows you the structure of the server response to a successful request.

Table 6-118 Plc_ReadCpuType_Response (object)

Name	Required	Data type	Description
product_name	Yes	string	The product name of the CPU or a CPU of the R/H system
order_number	Yes	string	The article number of the CPU or a CPU of the R/H system

Example

The following example shows the product name and the order number of the device.

```
{
  "product_name": "CPU 1516-3 PN/DP",
  "order_number": "6ES7 516-3AP03-0AB0"
}
```

6.17.2 Plc.ReadStationName

With this method you obtain the station name of the CPU.

No authorization is required to call the Plc.ReadStationName method.

Response structure

The following table shows you the structure of server responses to successful requests:

Table 6-119 Plc_ReadStationName_Response (object)

Name	Required	Data type	Description
station_name	Yes	string	The station name of the CPU

Example

In the following example, the user obtains the station name of the CPU.

```
{
  "station_name": "S71500/ET200MP station_1"
}
```

6.17.3 Plc.ReadModuleName

With this method you obtain the name of the CPU.

No authorization is required to call the Plc.ReadModuleName method.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 6-120 Plc_ReadModuleName_Request (object)

Name	Required	Data type	Description
redundancy_id	Yes, for R/H-CPUs; No for all other CPUs	number	The parameter "redundancy ID" must be available if the request is performed on an R/H-CPU. The "redundancy ID" has the value 1 or 2. This allows you to read out the name of both CPUs of an R/H system individually. With all other CPUs, the parameter must not be part of the request.

Response structure

The following table shows you the structure of server responses to successful requests:

Table 6-121 Plc_ReadModuleName_Response (object)

Name	Required	Data type	Description
module_name	Yes	string	The name of the CPU

Example

In the following example, the user obtains the article number of the CPU.

```
{
  "module_name": "PLC_1"
}
```

Possible error messages

The following table shows possible error messages of the Plc.ReadModuleName method.

Error code	Error message	Meaning
-32602	Invalid params	Invalid parameters of the method. This error occurs when the parameter redundancy_id is specified at a non-R/H-CPU or, respectively, when the parameter is invalid at an R/H-CPU.

6.18 Reading information from a redundant system

6.18.1 Redundancy.ReadSystemInformation

This method reads basic information on the redundant system and the pairing status of the R/H system S7-1500R/H.

To call the Redundancy.ReadSystemInformation method, you require "read_diagnostics" authorization.

Response structure

The following tables show the structure of server responses to successful requests.

Table 6-122 Redundancy_ReadSystemInformation_Response (object)

Name	Required	Data type	Description
pairing_state	Yes	string	Current pairing status of the redundant system S7-1500R/H Pairing is the mutual recognition of the two R/H-CPUs within a network. During pairing, the CPUs exchange information for mutual identification, for example checking for matching article number and firmware version.
syncup_lock	Yes	bool	The user may lock the SYNCUP or request the current SYNCUP status by calling the "RH_CTRL" instruction in the user program. Shows whether the SYNCUP is locked via the user program. One of the following states is possible: • true: The SYNCUP is currently blocked. The user cannot move the redundant system into the SYNCUP. • false: The SYNCUP is not blocked. The user can move the system in the SYNCUP.
standalone_operation	Yes	bool	The user can determine whether the CPU is running in single operation or not. Possible values are: • "true": The CPU works as an individual CPU (standalone) • "false": The CPU works in the normal redundancy mode of the R/H system. For firmware ≤ V3.1 the parameter is always "false".

Name	Required	Data type	Description
connected_redundancy_id	Yes	number	By assigning the redundancy IDs, you define for the R/H system which project data an R/H-CPU uses for itself. Provides the redundancy ID of the R/H-CPU via which the HTTP connection was established.
plcs	Yes	Object of type `Redundancy_ReadSystemInformation_Plcs_Response`	This object shows the redundant system and its two R/H-CPUs.

Table 6-123 Redundancy_ReadSystemInformation_Plcs_Response (object)

Name	Required	Data type	Description
plc_1	Yes	Object of type `Redundancy_ReadSystemInformation_Plc_Response`	This object shows the R/H-CPU with redundancy ID 1.
plc_2	Yes	Object of type `Redundancy_ReadSystemInformation_Plc_Response`	This object shows the R/H-CPU with redundancy ID 2.

Table 6-124 Redundancy_ReadSystemInformation_Plc_Response (object)

Name	Required	Data type	Description
redundancy_id	Yes	number	The redundancy ID of the displayed R/H-CPU: • For object plc_1, this attribute always contains the value 1. • For object plc_2, this attribute always contains the value 2.
role	yes	string	The role of the displayed R/H-CPU: The following roles are possible: • "primary": Primary CPU • "backup": Backup CPU • "unknown" unknown
hwid	Yes	number	The hardware ID of the displayed R/H-CPU: • For object plc_1 the hardware ID is always 65149. • For object plc_2 the hardware ID is always 65349.

More information

You can find detailed information on the design and mode of operation of S7-1500R/H in the Redundant system S7-1500R/H System Manual (https://support.industry.siemens.com/cs/ww/en/view/109754833).

Example 1

The following example shows a correctly configured R/H system and single-pairing access for CPU with redundancy ID 1.

```
{
  "pairing_state": "paired_single",
  "syncup_lock": false,
  "connected_redundancy_id": 1,
  "standalone_operation": false,
  "plcs":
  {
    "plc_1":
    {
      "redundancy_id": 1,
      "role": "backup",
      "hwid": 65149
    },
    "plc_2":
    {
      "redundancy_id": 2,
      "role": "primary",
      "hwid": 65349
    }
  }
}
```

Example 2

The following example shows a non-paired R/H system for CPU with redundancy ID 2.

```
{
  "pairing_state": "not_paired_firmware_mismatch",
  "syncup_lock": false,
  "connected_redundancy_id": 2,
  "standalone_operation": false,
  "plcs":
  {
    "plc_1":
    {
      "redundancy_id": 1,
      "role": "unknown",
      "hwid": 65149
    },
    "plc_2":
    {
      "redundancy_id": 2,
      "role": "primary",
```

```
      "hwid": 65349
    }
  }
}
```

Possible error messages

The following table shows possible error messages of the Redundancy.ReadSystemInformation method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.

6.18.2 Redundancy.ReadSystemState

This method reads the current system state of the redundant system S7-1500R/H.

To call the Redundancy.ReadSystemState method, you require "read_diagnostics" authorization.

Response structure

The following table shows you the structure of server responses to successful requests.

Table 6-125 Redundancy_ReadSystemState_Response (object)

Name	Required	Data type	Description
state	yes	string	Possible system states with your Web API designation: • STOP: stop • ANLAUF: startup • RUN-Solo: run_solo • SYNCUP: syncup • RUN-Redundant: run_redundant

More information

You can find detailed information on the design and mode of operation of S7-1500R/H in the Redundant system S7-1500R/H System Manual (https://support.industry.siemens.com/cs/ww/en/view/109754833).

Example

In the following example, the system state RUN-Redundant is read.

```
{
  "state": "run_redundant"
}
```

Possible error messages

The following table shows possible error messages of the Redundancy.ReadSystemState method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.

6.18.3 Redundancy.RequestChangeSystemState

With this method, you can change the system state of the redundant system S7-1500R/H.

To call the Redundancy.RequestChangeSystemState method, you require "change_operating_mode" authorization.

Structure of the request

The following table contains information about the parameters of the request:

Table 6-126 Redundancy_RequestChangeSystemState_Request (object)

Name	Required	Data type	Description
state	yes	string	The new system state Switching to the following system states is possible: • STOP: stop • RUN-Redundant: run_redundant

More information

You can find detailed information on the design and mode of operation of S7-1500R/H in the Redundant system S7-1500R/H System Manual (https://support.industry.siemens.com/cs/ww/en/view/109754833).

Example

In the following example, the new system state RUN-Redundant is read.

```
{
  "state": "run_redundant"
}
```

Possible error messages

The following table shows possible error messages of the Redundancy.RequestChangeSystemState method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.

6.18.4 Redundancy.ReadSyncupProgress

This method reads the information via a running SYNCUP of the redundant system S7-1500R/H. During the syncup, the web server can temporarily not be reached. Whether the method can be called or not thus depends on the current syncup phase and the duration of the syncup phase.

To call the Redundancy.ReadSyncupProgress method, you require "read_diagnostics" authorization.

Response structure

The following tables show the structure of server responses to successful requests.

Table 6-127 Redundancy_ReadSyncupProgress_Response (object)

Name	Required	Data type	Description
syncup_phase	Yes	string	Current SYNCUP phase of the redundant system S7-1500R/H. Possible phases are: • Copying the SIMATIC load memory (copying_memory_card) • Restarting of the backup CPU (rebooting_backup) • Finishing tasks (preparing_work_memory) • Copying the work memory (copying_work_memory) • Making up backup CPU lag (minimizing_delay)
copying_memory-_card	No	object of type `Redundancy_ReadSyncupProgress_CopyMemoryCard_Response`	If the "syncup_phase" parameter is in the "copying_memory_card" status, this object must be returned.
minimizing_delay	No	object of type `Redundancy_ReadSyncupProgress_MinimizingDelay_Response`	If the "syncup_phase" parameter is in the "minimizing_delay" status, this object must be returned.

Table 6-128 Redundancy_ReadSyncupProgress_MinimizingDelay_Response (object)

Name	Required	Data type	Description
hypothetical_cycle_time	Yes	string	ISO8601 time stamp as a string: The possible cycle duration in milliseconds
tolerable_cycle_time	Yes	string	ISO8601 time stamp as a string: The tolerable cycle duration in milliseconds

Table 6-129 Redundancy_ReadSyncupProgress_CopyMemoryCard_Response (object)

Name	Required	Data type	Description
current	Yes	number	The number of bytes from the SIMATIC load memory which was previously transferred to the backup CPU.
total	Yes	number	The overall number of the bytes to be processed of the SIMATIC load memory.

More information

You can find detailed information on the design and mode of operation of S7-1500R/H in the Redundant system S7-1500R/H System Manual (https://support.industry.siemens.com/cs/ww/en/view/109754833).

Example 1

In the following example, SYNCUP runs on the primary CPU: static content.

```
{
  "syncup_phase": "copying_memory_card",
  "copying_memory_card":
  {
   "current": 17024,
   "total": 2045000
  }
}
```

Example 2

In the following example, SYNCUP runs on the primary CPU: Making up backup CPU lag.

```
{
  "syncup_phase": "minimizing_delay",
  "minimizing_delay":
  {
   "hypothetical_time": "PT1S",
   "tolerable_time": "PT0.800S"
  }
}
```

Example 3

In the following example, SYNCUP runs on the primary CPU: Restart of the backup CPU.

```
{
  "syncup_phase": "rebooting_backup"
}
```

Example 4

In the following example, SYNCUP starts on the primary CPU:

```
{
  "syncup_phase": "checking_preconditions"
}
```

Example 5

In the following example, SYNCUP runs on the backup CPU.

```
{
  "syncup_phase": "in_progress"
}
```

Example 6

In the following example, no SYNCUP is running, the system is in the redundant state.

```
{
  "syncup_phase": "redundant"
}
```

Example 7

In the following example, no SYNCUP is running, the system is not in the redundant state.

```
{
  "syncup_phase": "not_redundant"
}
```

Possible error messages

The following table shows possible error messages of the Redundancy.ReadSyncupProgress method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.

Glossary

API

API (Application Programming Interface) is an interface through which various applications can communicate with each other with the purpose of exchanging data with each other.

ASCII

ASCII (American Standard Code for Information Interchange) is a US American standard for the coding of characters in computer systems, with each character being represented by a 7-bit binary number. The letter A is, for example, represented by the binary number (0)1000001. ASCII serves as the basis for subsequent encoding of character sets that are based on more bits.

Automation system

An automation system is a programmable logic control system, consisting of at least one CPU, diverse IO modules, and control and monitoring devices.

AWP commands

AWP (Automation Web Programming) commands are a special command syntax for exchanging data between a CPU and HTML file.

Configuration

Systematic arrangement of individual modules (design).

CPU

CPU (Central Processing Unit) is a processor that processes and controls the commands that are executed in a computer or electronic device. The processor is an electronic chip that functions as the brain of a computer or device.

CSS

A CSS (Cascading Style Sheet) specifies how an area or content marked up in HTML is displayed.

Device

Device that can send, receive or boost data via the bus, for example an IO Controller.

DHCP

DHCP (Dynamic Host Configuration Protocol) is a communication protocol for computer networks. In accordance with the client-server principle, it ensures that devices seeking

connections automatically obtain a reusable network address and all further relevant parameters.

DHCP server

A DHCP server automatically assigns IP addresses, gateways, and further network parameters within a network configuration to its clients.

Diagnostics

Diagnostics are the recognition, localization, classification, display and further evaluation of errors, disturbances and messages.

Diagnostics provides monitoring functions that run automatically while the system is in operation. This increases the availability of plants by reducing commissioning times and downtimes.

Firewall

A firewall is used to restrict the network access based on sender or target address of the used services. The firewall decides based on specified rules which of the network packets it handles are forwarded and which are not. This way the firewall tries to prevent unauthorized network access.

It is not the function of a firewall to detect attacks. It only implements rules for network communication.

Firmware

The firmware is a software that is embedded in electronic devices and performs fundamental functions there. It has an intermediate function between the hardware (meaning the physical components of a device) and the application software (the replaceable programs of a device). You must update the firmware regularly to, for example, fulfil the security requirements. The latest firmware versions are available in SiePortal.

GDS

A Global Discovery Server (GDS) makes mechanisms available for the central management of the following components:

- CA-signed certificates and self-signed certificates
- Trusted Lists and Certificate Revocation Lists (CRL)

HTTP

The Hypertext Transfer Protocol (HTTP) designates a protocol for transfer of data via a network. Transfer between server and client is effected unencrypted, meaning that all the information is transferred in plain text.

HTTPS

The Hypertext Transfer Protocol Secure (HTTPS) is a protocol for tap-proof transfer of sensitive data via a network. Transfer between server and client is effected encrypted.

Identification data

Identification data is stored on a module, and contains information which supports the user in

- Checking the system configuration
- Locating hardware changes in a system
- Correcting errors in a system

Modules can be clearly identified online using the identification data.

Industry Mall

The catalog and ordering system of Siemens AG for automation and drive solutions on the basis of Totally Integrated Automation (TIA) and Totally Integrated Power (TIP). Industry Mall is integrated together with SIOS in SiePortal.

JavaScript

Client-side script language that is mainly used to add interactive contents to web pages.

JSON

JSON (JavaScript Object Notation) is a compact data format in an easy-to-read text form that is often used in web applications to exchange data between servers and clients. JSON is platform-independent and is supported by many programming languages.

Master

The master in possession of the token is an active device. This master has the option to receive data from other devices and to send data to other devices.

Motion Control

Drives, modules and applications for controlling motion that encompass functions and components for coordination in terms of position and time for machine components in processing machinery.

OPC UA

Open Platform Communications Unified Architecture is an open, Ethernet-based communication standard between machines and vertically into the Cloud. Siemens OPC UA is combined with PROFINET in a shared Industrial Ethernet network.

PROFINET

Within the context of Totally Integrated Automation (TIA), PROFINET is the systematic continuation of:

- PROFIBUS DP, the established fieldbus
- Industrial Ethernet, the communications bus for the cell level

Experiences from both systems have been are integrated in PROFINET.

PROFINET as an Ethernet-based automation standard of PROFIBUS International (formerly PROFIBUS User Organization e.V.) thus defines a manufacturer-independent communication, automation and engineering model.

Technology objects

Especially in the field of "Motion Control" the technology objects (TO) serve to simplify the actuation and handling of axes and additional Motion Control functionalities and thus support the user in creating a user program with Motion Control functionality.

The technology object

- Represents a software object in the PLC:
- Represents the mechanical components
- Encapsulates the technological functionality
- Allows uniform configuration and parameter assignment
- Ensures simple connection of the drives, encoders and I/O devices
- Encapsulates the mechanical structure, the monitoring functions and the limitations of the drive and the mechanical components connected to them

The technology object is addressed via PLCopen Motion Control instructions from the user program. This ensures simple and standardized application of the Motion Control functionalities in SIMATIC.

TIA

Totally Integrated Automation The organization concept of the manufacturer Siemens. This concept defines the interaction of different individual automation technology components, software tools and associated services (spare part service, etc.) into a consistent automation solution.

TIP

Totally Integrated Power The integrated power supply solutions for industries, buildings and infrastructure facilities of the manufacturer Siemens. The comprehensive power supply portfolio enables reliable, safe and efficient power supply with software and hardware products, systems and solutions for all voltage levels.

URL

The Uniform Resource Locator identifies and localizes a source, such as a web page, uniquely via the method of access used and the location of the source in computer networks.

UTC

Coordinated Universal Time. The Coordinated Universal Time is the world time introduced in 1972 and valid today.

UTF-8

UTF-8 is the abbreviation for 8-bit UCS (Universal Character Set) Transformation Format. UTF-8 is the most widely used encoding of Unicode characters.

Each Unicode character is assigned a specially encoded byte string of variable length. UTF-8 supports up to 4 bytes onto which all Unicode characters can be mapped.

VoT

View of Things Web-based visualizations for simple operator control and monitoring of SIMATIC devices.

Web API

A Web API is a type of web service that makes it possible to make available and use data and functionalities via the internet. Web API allows the applications to communicate with each other and exchange data by making an interface available that allows applications to access specific resources and to manipulate these.

Index

A

Access restriction, 25
Activating the Web server, 22
API, 71
Application name, 120
Arrays, 156

B

Binary representation, 152

C

CA certificate, 30
Certificate
  Web server certificate, 30
Certificate manager
  Global security settings, 31
  Global CA-signed certificate, 31
  Local CPU-specific certificate, 35
CPU-specific certificate, 30

D

Display of texts in different languages, 40
Download, 103

E

End point
  API, 77
  Ticket, 103
  Web applications that can be loaded by the user, 119

F

FAQs
  Web server access via smartphone, 21
  Download certificate, 30

G

Github, 78
Global security settings, 31

H

HTTPS, 23

J

JSON-RPC, 77, 102

L

Language settings, 39
Logging out, 55
Login, 53

M

Media type
  Name, 120

O

Overview
  Login, 53

P

Password, 56

R

Resource name, 120

S

Security functions, 19
Self-signed certificates, 30

T

Ticket end point, 103
Ticket ID, 103
Ticket mechanism, 102
Ticket methods, 105

U

Upload, 103
User-defined web applications, 117
User interface language
  assigning to project language, 40

V

Version number, 120

W

Web access
  Via PG/PC, 21
  Via HMI devices and mobile terminal devices, 21
Web API, 71
  supported clients, 78
  Web applications, 117
  supported data types, 152
Web application end point, 77
Web applications, 117
Web applications that can be loaded by the user, 117
Web browser, 19
Web server
  Properties, 19
  Certificate, 30
Web server certificate
  creating and assigning, 30
Web server project language, 25