User Manual

Armatura Phalanx Protocol

Armatura Phalanx Protocol 1.0

Date: May 2026

Doc Version: 1.0

User Manual

Armatura Phalanx Protocol

Phalanx Protocol SDK (Controller Webserver API)

PSIA-Compliant RESTful API for AHSC-1000 Controller and AHDU Series Controller

Document Information	
Product	Armatura Phalanx Protocol SDK
API Type	RESTful Web API (HTTP/HTTPS)
Data Format	XML (per PSIA schemas)
Supported Products	AHSC-1000 Controller, AHDU Series Controller
SDK Version	1.0.0
Document Version	1.0
Date	May 2026
Document Code	ArmaSec-04232026

Copyright

Trademark

ARMATURA and the ARMATURA logo are trademarks of Armatura LLC. Other trademarks involved in this manual are owned by their respective owners.

Disclaimer

This manual contains information on the operation, integration, and maintenance of the ARMATURA Phalanx Protocol SDK. The copyright in all documents, drawings, specifications, and software-related information in relation to the ARMATURA supplied equipment vests in and is the property of ARMATURA. The contents hereof should not be used or shared by the receiver with any third party without express written permission of ARMATURA.

The contents of this manual must be read before starting integration, operation, or maintenance of the supplied equipment. If any content of this manual seems unclear or incomplete, please contact ARMATURA before starting the operation, integration, or maintenance of the said equipment.

In case of any conflict between terms and conditions of this manual and the contract specifications, drawings, instruction sheets, product datasheets, or any other contract-related documents, the contract conditions and final product datasheets shall prevail.

ARMATURA offers no warranty, guarantee, or representation regarding the completeness of any information contained in this manual or any amendments made thereto. ARMATURA does not extend any warranty of design, merchantability, or fitness for a particular purpose.

ARMATURA in no event shall be liable to the user or any third party for any incidental, consequential, indirect, special, or exemplary damages, including, without limitation, loss of business, loss of profits, business interruption, loss of business information, or any pecuniary loss arising out of, in connection with, or relating to the use of the information contained in or referenced by this manual, even if ARMATURA has been advised of the possibility of such damages.

The product and protocol will be updated from time to time without prior notice. The latest operation procedures and relevant documents are available from ARMATURA.

ARMATURA Headquarters

Address	190 Bluegrass Valley Parkway Alpharetta, GA 30005
Phone	+1 (470) 816-1970
Email	sales@armatura.us
Website	www.armatura.us

About the Manual

This manual introduces the Armatura Phalanx Protocol SDK, the embedded controller webserver API for integration with Armatura AHSC-1000 Controller and AHDU Series Controller products. It explains the protocol scope, deployment model, security mechanism, API modules, representative XML request examples, and recommended integration workflows.

All figures and examples are provided for illustration and integration reference. API availability, field definitions, and payload requirements may vary by firmware version, controller configuration, license, and project-specific SDK customization. Verify the live Swagger UI and PSIA XSD schemas on the target device during project implementation.

Revision History

Version	Date	Description	Author
1.0	May 2026	Initial official user manual draft based on the Phalanx Protocol Swagger specification, the final product leaflet/datasheet, and the Armatura regular user manual layout.	ARMATURA

Table of Contents

1. Product Overview

1.1 Introduction

1.2 Supported Products and Standards

1.3 Deployment Architecture

1.4 Function List

2. Getting Started

2.1 Preconditions

2.2 Base URL, Transport, and Data Format

2.3 Authentication Context

2.4 Request and Response Conventions

3. System API

4. Authentication and Device Ownership API

5. Access Control API

6. Metadata Event Session API

7. Integration Workflows

8. Cybersecurity and Data Protection

9. Troubleshooting

Appendix A. Full API Endpoint Summary

Appendix B. XML Request Examples

Appendix C. Terminology

1. Product Overview

1.1 Introduction

Armatura Phalanx Protocol SDK is Armatura's official implementation of PSIA-compliant access control interoperability for controller webserver integration. It is designed as a modern, developer-first RESTful API that enables third-party software platforms to communicate directly with Armatura intelligent access control controllers using standard HTTP/HTTPS endpoints and XML payloads defined by PSIA schemas.

The SDK supports integration scenarios involving Access Control Systems (ACS), Video Management Systems (VMS), Parking Systems, Building Management Systems (BMS), Physical Security Information Management (PSIM) platforms, enterprise software solutions, and custom-developed applications. It eliminates the need for proprietary drivers, middleware, or complex integration layers when a project requires direct controller-level communication.

The protocol scope includes credential and credential-holder management, time schedules, holidays, permissions, roles, door/portal objects, controller status, system time, NTP server configuration, device ownership, authorized users, and real-time event sessions.

1.2 Supported Products and Standards

General Information	
Supported Products	AHSC-1000 Controller, AHDU Series Controller
Protocol	Armatura Phalanx Protocol (PSIA/PLAI compliant)
API Type	RESTful Web API (HTTP/HTTPS)
Data Format	XML (per PSIA schemas)
Documentation	Full Swagger UI + PSIA XSD schemas
Standards	PSIA Area Control 3.1, PLAI, CSEC, Common Metadata Event
Authentication	Basic Auth, Issuer-Signature, CSEC Device Ownership
Event Streaming	PSIA Metadata Event sessions; TCP/UDP/raw data support may depend on firmware and deployment configuration

Product naming: This manual uses the official product naming AHSC-1000 Controller and AHDU Series Controller. Internal engineering names are not used in the user-facing manual.

1.3 Deployment Architecture

In a typical deployment, the embedded web server of the Armatura controller acts as the system server. Third-party systems connect to the controller through the Phalanx API over TCP/IP using HTTPS and TLS encryption. The third-party system sends RESTful requests to the controller, while event sessions can be established for real-time event and alarm monitoring.

1.4 Function List

Functions	Operation Instructions
System	View device profile and system status; read device time; create, update, query, and delete NTP server settings.
Authentication	Manage device ownership status and authorized users through CSEC device ownership, owner-code, and issuer-signature controls.
Access Control - Basic	Query controller configuration and current access control status.
Wiegand Formats	Create, read, update, and delete Wiegand card format definitions used for card-number interpretation.
Permissions	Create, read, update, and delete permission objects linking portals/doors to time schedules.
Credential Holders	Create, read, update, and delete personnel or credential-holder records.
Roles	Query, update, and delete access roles, including role-to-permission associations.
Credentials	Create, read, update, and delete credentials such as card identifiers and PIN values assigned to credential holders.

Functions	Operation Instructions
Doors/Portals	Retrieve the list of portal/door objects available on the AHSC-1000 Controller and AHDU Series Controller environment.
Holidays	Create, read, update, and delete holiday calendars used by access-control schedules.
Time Schedules	Create, read, update, and delete access-control time schedules, including regular day and holiday intervals.
Metadata Event Sessions	Query session support; create synchronous or asynchronous event streams; start, read, and delete event sessions.

2. Getting Started

2.1 Preconditions for Normal Use of Functions

- The AHSC-1000 Controller or AHDU Series Controller is installed, powered, and connected to the network.
- The third-party system can reach the controller IP address or domain name over the project network.
- HTTPS is enabled and the deployment is configured to use TLS encryption according to the project security policy.
- The integrator has valid authorization credentials, including the required Basic Auth, CSEC ownership, owner-code, and issuer-signature information for protected resources.
- The integrator has access to the target controller Swagger UI and PSIA XSD schemas for firmware-specific payload validation.
- Controller time, NTP settings, and time zone configuration are verified before access-control and event-session testing.

2.2 Base URL, Transport, and Data Format

Request Convention	
Base URL Format	https://<controller-ip-or-domain>
Transport	HTTPS over TCP/IP; TLS 1.2+ recommended/required by deployment policy
Content-Type	application/xml
Accept	application/xml
Payload Format	XML documents using the namespace urn:psialliance-org and PSIA-defined structures
OpenAPI Version	OAS 3.0.1

Example request header pattern

```
GET https://192.168.1.100/System/status
Accept: application/xml
Cookie: owner-code=<ownerCode>; issuer-signature=<issuerSignature>
```

2.3 Authentication Context

The Phalanx Protocol supports an advanced security model that combines Basic Authentication, CSEC device ownership, issuer-signature authentication, owner-code control, and TLS-protected sessions. Most protected endpoints require owner-code and issuer-signature values to be sent in request cookies. Certain authentication and ownership operations also use query parameters and headers defined by the Swagger specification.

Authentication and Security Parameters	
Basic Auth	HTTP Basic Authentication scheme defined in the OpenAPI specification.
Device Ownership	CSEC ownership model used to establish and control trusted ownership of the device node.
owner-code	Cookie value used by protected operations after ownership/authorization has been established.
issuer-signature	Cookie value used by protected operations to verify request issuer authorization.
IssuerSignature	Header value used by selected CSEC operations, depending on endpoint definition.

TLS	Encrypts API traffic and helps protect credentials, signatures, and access-control data during transport.

Security: Do not use example passwords, GUIDs, owner codes, or signatures in production. Replace all sample values with project-specific values generated by the target device and integration workflow.

2.4 Request and Response Conventions

- GET operations are used to query lists, individual resources, device status, configuration, or event session parameters.
- POST operations are used to create resources such as NTP servers, authorized users, Wiegand formats, permissions, credential holders, credentials, holidays, time schedules, and metadata event sessions.
- PUT operations are used to update full resource objects or resource lists, depending on the endpoint.
- DELETE operations are used to remove resources or clear all resources for supported collection endpoints.
- Path parameters enclosed in braces, such as {id} or {credentialUID}, must be replaced by valid IDs or GUIDs from the target controller.
- The Swagger specification represents many response schemas as generic XML objects. Verify exact response fields on the target device Swagger UI and PSIA XSD files during implementation.

3. System API

The System API provides read-only device profile/status operations and time management functions. Integrators should verify controller status and time synchronization before testing access-control resources or event sessions.

3.1 Function Description

- Query the device node information through the profile endpoint.
- Query the current system status of the controller.
- Read device system time.
- Configure one or multiple NTP servers for system time synchronization.

3.2 System API Reference

Method	Endpoint	Operation
GET	`/System/time/ntpServers`	Get NTP server list
POST	`/System/time/ntpServers`	Create a NTP server
DELETE	`/System/time/ntpServers`	Delete all NTP server
PUT	`/System/time/ntpServers/{id}`	Update a NTP server
GET	`/System/time/ntpServers/{id}`	Get a NTP server
DELETE	`/System/time/ntpServers/{id}`	Delete a NTP server
GET	`/System/time`	Get device system time
GET	`/profile`	Get device node information
GET	`/System/status`	Get device system status

3.3 Example: Create NTP Server

```
<?xml version="1.0" encoding="UTF-8"?>
<NTPServer xmlns="urn:psialliance-org" version="1.0">
  <id>5</id>
  <addressingFormatType>ipaddress</addressingFormatType>
  <ipAddress>120.25.115.20</ipAddress>
</NTPServer>
```

3.4 Example: Update NTP Server

```
<?xml version="1.0" encoding="UTF-8"?>
<NTPServer xmlns="urn:psialliance-org" version="1.0">
  <id>5</id>
  <addressingFormatType>ipaddress</addressingFormatType>
  <ipv6Address>2001:da8:9000::81</ipv6Address>
</NTPServer>
```

4. Authentication and Device Ownership API

The Authentication and CSEC APIs support device ownership query/reset and authorized user management. These functions should be restricted to trusted administrators and integration services because they directly affect the trust model used by protected resources.

4.1 Device Ownership

Device ownership identifies whether the controller is owned and trusted by a configured issuer/owner. Ownership information is queried before protected operations are executed. The reset or deletion of ownership should only be performed during controlled commissioning, re-commissioning, or service recovery procedures.

4.2 Authorized Users

Authorized user endpoints are used to query, create, update, and delete CSEC authorized users. Users may be addressed by numeric ID or GUID depending on the operation.

4.3 Authentication API Reference

Method	Endpoint	Operation
GET	/CSEC/deviceOwnership	Get device node authorization
DELETE	/CSEC/deviceOwnership	Delete device node authorization
GET	/CSEC/AAA/users	Get authorized user list
POST	/CSEC/AAA/users	Create an authorized user
PUT	/CSEC/AAA/users	Update authorized user list
PUT	/CSEC/AAA/users/{id}	Update an authorized user (by id)
DELETE	/CSEC/AAA/users/{id}	Delete an authorized user (by id)
GET	/CSEC/AAA/users/{id}	Get an authorized user (by id)
PUT	/CSEC/AAA/users/{GUID}	Update an authorized user (by GUID)
DELETE	/CSEC/AAA/users/{GUID}	Delete an authorized user (by GUID)
GET	/CSEC/AAA/users/{GUID}	Get an authorized user (by GUID)

4.4 Example: Create Authorized User

```
<?xml version="1.0" encoding="UTF-8"?>
<CSECUser xmlns="urn:psialliance-org" version="1.0">
    <id>3</id>
    <userEnabled>true</userEnabled>
    <userGUID>{E6152A7C-41F8-8006-2728-0A1FB495FF46}</userGUID>
    <UserLogin>
        <username>admin</username>
        <password>21232f297a57a5a743894a0e4a801fc3</password>
    </UserLogin>
</CSECUser>
```

4.5 Example: Update Authorized User by ID

```
<?xml version="1.0" encoding="UTF-8"?>
<CSECUser xmlns="urn:psialliance-org" version="1.0">
    <id>3</id>
    <userEnabled>false</userEnabled>
    <userGUID>{E6152A7C-41F8-8006-2728-0A1FB495FF46}</userGUID>
    <UserLogin>
        <username>admin</username>
        <password>21232f297a57a5a743894a0e4a801fc3</password>
    </UserLogin>
</CSECUser>
```

Implementation note: The password values in the sample XML appear as hashed strings. Follow the product-specific password hashing and authorization guidance supplied for the target firmware and project security requirements.

5. Access Control API

The Access Control API is the main API group for managing access-control business objects. It includes configuration/status queries, Wiegand format definitions, permission objects, credential holders, roles, credentials, door/portal lists, holidays, and time schedules.

5.1 Access Control Data Relationship

A typical access-control data model follows this relationship: a credential holder represents a person; credentials such as card and PIN values are assigned to the credential holder; roles and permissions define which doors/portals may be accessed during defined time schedules and holiday intervals.

Credential Holder	→	Credential	→	Role	→	Permission = Portal/Door + Time Schedule + Holiday Rule

5.2 Access Control Module Description

Access Control Resources	
Basic	Query access-control configuration and status for the controller.
Wiegand formats	Define Wiegand card format length, data fields, facility code, card number fields, and parity fields.
Permissions	Define access privileges by binding time schedules and portal/door objects.
Credential holders	Manage personnel identity records used by access-control credentials.
Roles	Organize permissions into role objects that can be assigned to credential holders.
Credentials	Create card, PIN, and other credential identifiers and assign them to credential holders.
Doors/Portals	Query available portal/door resources exposed by the controller.
Holidays	Maintain holiday definitions used by time schedules.
Time schedules	Define regular and holiday-based time intervals for access-control rules.

5.3 Access Control API Reference

Method	Endpoint	Operation
GET	`/AreaControl/configuration`	Get configuration information
GET	`/AreaControl/status`	Get status information
GET	`/AreaControl/CredentialFormats/info`	Get Wiegand format list
POST	`/AreaControl/CredentialFormats/info`	Create a Wiegand format
GET	`/AreaControl/CredentialFormats/info/{formatID}`	Get a Wiegand format
PUT	`/AreaControl/CredentialFormats/info/{formatID}`	Update a Wiegand format
DELETE	`/AreaControl/CredentialFormats/info/{formatID}`	Delete a Wiegand format
GET	`/AreaControl/Permissions/info`	Get credential permission list
POST	`/AreaControl/Permissions/info`	Create a credential permission
PUT	`/AreaControl/Permissions/info/{permissionID}`	Update a credential permission
DELETE	`/AreaControl/Permissions/info/{permissionID}`	Delete a credential permission
GET	`/AreaControl/Permissions/info/{permissionID}`	Get a credential permission
GET	`/AreaControl/CredentialHolders/info`	Get credential holder list
POST	`/AreaControl/CredentialHolders/info`	Create a credential holder
GET	`/AreaControl/CredentialHolders/info/{credentialHolderID}`	Get a credential holder
DELETE	`/AreaControl/CredentialHolders/info/{credentialHolderID}`	Delete a credential holder
PUT	`/AreaControl/CredentialHolders/info/{credentialHolderID}`	Update a credential holder
GET	`/AreaControl/Roles/info`	Get role list
GET	`/AreaControl/Roles/info/{roleUID}`	Get a role
PUT	`/AreaControl/Roles/info/{roleUID}`	Update a role
DELETE	`/AreaControl/Roles/info/{roleUID}`	Delete a role
GET	`/AreaControl/Credentials/info`	Get credential list
POST	`/AreaControl/Credentials/info`	Create a credential
GET	`/AreaControl/Credentials/info/{credentialUID}`	Get a credential
PUT	`/AreaControl/Credentials/info/{credentialUID}`	Update a credential
DELETE	`/AreaControl/Credentials/info/{credentialUID}`	Delete a credential
GET	`/AreaControl/PartitionMembers/Portals/info`	Get door list
GET	`/AreaControl/Holidays/info`	Get Holiday List
POST	`/AreaControl/Holidays/info`	Create a Holiday
GET	`/AreaControl/Holidays/info/{HolidayID}`	Get a Holiday
PUT	`/AreaControl/Holidays/info/{HolidayID}`	Update a Holiday

Method	Endpoint	Operation
DELETE	/AreaControl/Holidays/info/{HolidayID}	Delete a Holiday
GET	/AreaControl/TimeSchedules/info	Get TimeSchedule List
POST	/AreaControl/TimeSchedules/info	Create a TimeSchedule
GET	/AreaControl/TimeSchedules/info/{TimeScheduleID}	Get a TimeSchedule
PUT	/AreaControl/TimeSchedules/info/{TimeScheduleID}	Update a TimeSchedule
DELETE	/AreaControl/TimeSchedules/info/{TimeScheduleID}	Delete a TimeSchedule

5.4 Example: Create Credential Holder

```
<?xml version="1.0" encoding="UTF-8"?>
<CredentialHolderInfo version="1.0" xmlns="urn:psialliance-org">
  <ID>1</ID>
  <UID>{82E01D49-8D02-4FA5-8AB3-1302C8483DFE}</UID>
  <Name>Last456,FirstName456</Name>
  <GivenName>Last456</GivenName>
  <Surname>FirstName456</Surname>
  <Email>Last456@mail.com</Email>
  <CreationDate>2024-05-14T13:48:56</CreationDate>
  <ActiveFrom>2024-05-30T10:55:00</ActiveFrom>
  <ActiveTill>2024-12-30T10:55:00</ActiveTill>
  <State>Active</State>
  <RoleIDList>
    <RoleID>
      <ID>1</ID>
      <GUID>{A7F15B05-A443-F1EB-76B0-509663334711}</GUID>
      <Name>System Administration</Name>
    </RoleID>
  </RoleIDList>
</CredentialHolderInfo>
```

5.5 Example: Create Credential

```
<?xml version="1.0" encoding="UTF-8"?>
<CredentialInfo xmlns="urn:psialliance-org" version="1.0">
    <ID>1</ID>
    <UID>{abacab00-0000-1761-100a-761234500001}</UID>
    <Name>Credential 1</Name>
    <AssignedToID>
        <ID>1</ID>
        <GUID>{82E01D49-8D02-4FA5-8AB3-1282C8483DDE}</GUID>
    </AssignedToID>
    <State>Active</State>
    <LastModifiedDate>2024-10-10T11:05:10</LastModifiedDate>
    <IdentifierInfoList>
        <IdentifierInfo>
            <Type>Card</Type>
            <Value>1349537517</Value>
            <ValueEncoding>Decimal</ValueEncoding>
        </IdentifierInfo>
        <IdentifierInfo>
            <Type>PIN</Type>
            <Value>666666</Value>
        </IdentifierInfo>
    </IdentifierInfoList>
</CredentialInfo>
```

5.6 Example: Create Credential Permission

```
<?xml version="1.0" encoding="UTF-8"?>
<PermissionInfo xmlns="urn:psialliance-org" version="1.0">
  <ID>2</ID>
  <UID>{65DF0744-AE13-F22D-51C2-F44FD1D4AC55}</UID>
  <Name>test</Name>
  <PrivilegeList>
    <Privilege>
      <Allow>
        <TimeScheduleIDList>
          <TimeScheduleID>
            <ID>1</ID>
            <GUID>{55B0071D-4BAD-FE9A-A263-E39D5BAB9DC7}</GUID>
            <Name>24-Hour Accessible</Name>
          </TimeScheduleID>
        </TimeScheduleIDList>
        <PortalIDList>
          <PortalID>
            <ID>1</ID>
            <GUID>{D3B66429-9A99-49AD-8407-851F2B4C291A}</GUID>
```



```
            <Name>10.8.12.234-1</Name>
          </PortalID>
          <PortalID>
            <ID>2</ID>
            <GUID>{6A201845-F8A4-4905-9413-182022D48E34}</GUID>
            <Name>10.8.12.234-2</Name>
          </PortalID>
        </PortalIDList>
      </Allow>
    </Privilege>
  </PrivilegeList>
</PermissionInfo>
```

5.7 Example: Create Holiday

```
<?xml version="1.0" encoding="UTF-8"?>
<HolidayInfo xmlns="urn:psialliance-org" version="1.0">
    <ID>7</ID>
    <UID>{62BC68F0-5F6F-49B0-ACC1-00001234500F}</UID>
    <Name>New Years Day</Name>
    <Description>January 1</Description>
    <RecursYearly>true</RecursYearly>
    <StartDate>2025-01-01</StartDate>
    <EndDate>2025-01-03</EndDate>
</HolidayInfo>
```

5.8 Example: Create Time Schedule

```
<?xml version="1.0" encoding="UTF-8"?>
<TimeScheduleInfo xmlns="urn:psialliance-org" version="1.0">
  <ID>2</ID>
  <UID>{76A628E1-EDF6-EAEC-7570-286E8B500D7E}</UID>
  <Name>8-Hour Accessible</Name>
  <Description>8-Hour Accessible</Description>
  <TimeIntervalInfoList>
    <TimeIntervalInfo>
      <Day>Monday</Day>
      <StartTime>09:00:00</StartTime>
      <EndTime>16:59:00</EndTime>
    </TimeIntervalInfo>
    <TimeIntervalInfo>
      <Day>Holiday</Day>
      <HolidayID>
        <ID>7</ID>
        <GUID>{62BC68F0-5F6F-49B0-ACC1-00001234500F}</GUID>
        <Name>New Years Day</Name>
      </HolidayID>
      <StartTime>10:00:00</StartTime>
      <EndTime>18:00:00</EndTime>
    </TimeIntervalInfo>
  </TimeIntervalInfoList>
</TimeScheduleInfo>
```

Dependency order: When creating access-control objects, create or verify dependent objects first. For example, verify portals/doors and holidays before creating permission and time-schedule objects that reference them.

6. Metadata Event Session API

The Metadata Event Session API supports real-time access events, alarms, and system status monitoring through PSIA Metadata Event sessions. Integrators use this API to create, query, start, and delete event streams based on the session type and workflow defined by the device.

6.1 Event Session Concepts

Event Session Terms	
sessionSupport	Queries the device node session parameters supported by the controller.
default synchronization session	Creates a default synchronization session through the GET /Metadata/stream operation.
event session	Creates a configured event session through POST /Metadata/stream.
synchronous session	GET /Metadata/stream/{streamID} may start the session when the stream is synchronous.

asynchronous session	GET /Metadata/stream/{streamID} may return session parameters when the stream is asynchronous.

6.2 Metadata Event Session API Reference

Method	Endpoint	Operation
GET	`/Metadata/sessionSupport`	Get device node session parameters
GET	`/Metadata/stream`	Create a default synchronization session
POST	`/Metadata/stream`	Create an event session
DELETE	`/Metadata/stream`	Delete all event sessions
GET	`/Metadata/stream/{streamID}`	Get an event session (start session if synchronous, return parameters if asynchronous)
DELETE	`/Metadata/stream/{streamID}`	Delete an event session

6.3 Example: Create Event Session

```
<?xml version="1.0" encoding="UTF-8"?>
<MetaSessionParms xmlns="urn:psialliance-org" version="1.1">
    <metaXportParms>
        <metaSessionID>0</metaSessionID>
        <metaFormat>xml-psia</metaFormat>
        <metaSessionType>RESTSyncSessionTargetSend</metaSessionType>
        <metaSessionFlowType>datastream</metaSessionFlowType>
    </metaXportParms>
</MetaSessionParms>
```

7. Integration Workflows

This section provides recommended workflows for project implementation and testing. The exact sequence may vary depending on controller firmware, deployment architecture, ownership status, and third-party platform requirements.

7.1 Device Onboarding and API Readiness

Step	Operation	Description
1	**Confirm network reachability**	Ping or otherwise confirm IP/domain reachability between the third-party system and the controller network.
2	**Open controller API endpoint**	Access the controller webserver using HTTPS and verify that the live Swagger UI is available.
3	**Validate device profile**	Call GET /profile and GET /System/status to confirm the target controller identity and status.
4	**Verify time synchronization**	Call GET /System/time and configure NTP servers if the controller time is not aligned with the project time source.
5	**Establish security context**	Complete Basic Auth, CSEC ownership, owner-code, and issuer-signature setup according to the project security policy.

7.2 Personnel and Credential Provisioning

Step	Operation	Description
1	**Query doors and schedules**	Query portal/door resources and time schedules needed by the target access policy.
2	**Create or update credential holder**	Use the credential-holder API to create or maintain the person record.
3	**Create credential**	Assign card/PIN/other credential identifiers to the credential holder.
4	**Create or update permission**	Bind permitted portals/doors to valid time schedules and holiday rules.
5	**Assign role/permission**	Use role or permission references according to the data model required by the project.
6	**Test access event**	Perform a controlled access event and confirm that the controller and third-party platform receive the correct result.

7.3 Event Session Monitoring

Step	Operation	Description
1	Query session support	Call GET /Metadata/sessionSupport to confirm supported session parameters.
2	Create event session	Call POST /Metadata/stream with the required MetaSessionParms payload.
3	Start or query session	Call GET /Metadata/stream/{streamID} according to synchronous or asynchronous session behavior.
4	Consume events	Parse PSIA metadata events from the event stream and map them to the third-party platform event model.
5	Close session	Call DELETE /Metadata/stream/{streamID} or DELETE /Metadata/stream when monitoring is complete.

8. Cybersecurity and Data Protection

The Phalanx Protocol is designed for secure integration, but project security also depends on network architecture, credential handling, operational policy, and correct configuration. The following practices should be applied during system deployment and maintenance.

- Use HTTPS and TLS 1.2+ for all API communication. Do not transmit credentials, owner codes, issuer signatures, or access-control payloads over unencrypted channels.
- Change default or commissioning passwords before production use and assign unique administrator/integration accounts per project policy.
- Restrict Phalanx API access to trusted network segments, VPNs, firewalls, or zero-trust access policies as required by the project.
- Store owner-code, issuer-signature, Basic Auth credentials, and certificates in an encrypted secret store. Never hardcode production secrets in source code.
- Grant only the permissions needed by the integration service. Separate commissioning, maintenance, monitoring, and daily operation accounts where possible.
- Log API access and monitor abnormal request rates, repeated authentication failures, unusual DELETE operations, and unexpected ownership changes.
- Validate XML inputs and outputs against the relevant PSIA XSD schemas and firmware-specific Swagger UI.
- Protect personally identifiable information, credential identifiers, and access events according to applicable privacy laws and project policies.

Data Protection Summary	
Communication Security	HTTPS, TLS encryption
Authentication	Issuer-Signature, CSEC permission model, Device Ownership
Data Protection	Secure session management, encrypted credential handling
Standards	PSIA Area Control, CSEC, Metadata Event

9. Troubleshooting

Symptom	Recommended Action
Cannot access Swagger UI	Verify controller IP/domain, routing, firewall policy, HTTPS port, and controller webserver status.
Authentication failure	Verify Basic Auth credentials, CSEC ownership state, owner-code, issuer-signature, and required headers/cookies for the endpoint.
XML payload rejected	Validate namespace, root element, required IDs/GUIDs, date/time formats, and dependencies such as portal IDs or schedule IDs.
NTP/time-related errors	Query controller system time, verify NTP server configuration, and confirm network reachability to the time server.
Permission does not take effect	Confirm that the credential holder, credential, role, permission, portal/door, holiday, and time schedule references are all valid and active.
No real-time events received	Verify Metadata session support, create/start the correct stream, check stream ID, and confirm network path for the event session.
DELETE operation has unexpected result	Confirm whether the endpoint deletes one resource or all resources in the collection before executing the command.

Appendix A. Full API Endpoint Summary

The following endpoint summary is generated from the embedded OpenAPI 3.0.1 Swagger specification supplied with the Armatura Phalanx Protocol SDK. Confirm exact behavior on the target controller firmware during implementation.

Module	Method	Endpoint	Operation
System/Time/NTP Server	GET	/System/time/ntpServers	Get NTP server list
System/Time/NTP Server	POST	/System/time/ntpServers	Create a NTP server
System/Time/NTP Server	DELETE	/System/time/ntpServers	Delete all NTP server
System/Time/NTP Server	PUT	/System/time/ntpServers/{id}	Update a NTP server
System/Time/NTP Server	GET	/System/time/ntpServers/{id}	Get a NTP server
System/Time/NTP Server	DELETE	/System/time/ntpServers/{id}	Delete a NTP server
System/Time	GET	/System/time	Get device system time
System	GET	/profile	Get device node information
System	GET	/System/status	Get device system status
Device Ownership	GET	/CSEC/deviceOwnership	Get device node authorization
Device Ownership	DELETE	/CSEC/deviceOwnership	Delete device node authorization
Authorized users	GET	/CSEC/AAA/users	Get authorized user list
Authorized users	POST	/CSEC/AAA/users	Create an authorized user
Authorized users	PUT	/CSEC/AAA/users	Update authorized user list
Authorized users	PUT	/CSEC/AAA/users/{id}	Update an authorized user (by id)
Authorized users	DELETE	/CSEC/AAA/users/{id}	Delete an authorized user (by id)
Authorized users	GET	/CSEC/AAA/users/{id}	Get an authorized user (by id)
Authorized users	PUT	/CSEC/AAA/users/{GUID}	Update an authorized user (by GUID)
Authorized users	DELETE	/CSEC/AAA/users/{GUID}	Delete an authorized user (by GUID)
Authorized users	GET	/CSEC/AAA/users/{GUID}	Get an authorized user (by GUID)
Basic	GET	/AreaControl/configuration	Get configuration information
Basic	GET	/AreaControl/status	Get status information
Wiegand formats	GET	/AreaControl/CredentialFormats/info	Get Wiegand format list
Wiegand formats	POST	/AreaControl/CredentialFormats/info	Create a Wiegand format
Wiegand formats	GET	/AreaControl/CredentialFormats/info/{formatID}	Get a Wiegand format
Wiegand formats	PUT	/AreaControl/CredentialFormats/info/{formatID}	Update a Wiegand format
Wiegand formats	DELETE	/AreaControl/CredentialFormats/info/{formatID}	Delete a Wiegand format
Permissions	GET	/AreaControl/Permissions/info	Get credential permission list
Permissions	POST	/AreaControl/Permissions/info	Create a credential permission
Permissions	PUT	/AreaControl/Permissions/info/{permissionID}	Update a credential permission
Permissions	DELETE	/AreaControl/Permissions/info/{permissionID}	Delete a credential permission
Permissions	GET	/AreaControl/Permissions/info/{permissionID}	Get a credential permission
Credential holders	GET	/AreaControl/CredentialHolders/info	Get credential holder list
Credential holders	POST	/AreaControl/CredentialHolders/info	Create a credential holder
Credential holders	GET	/AreaControl/CredentialHolders/info/{credentialHolderID}	Get a credential holder
Credential holders	DELETE	/AreaControl/CredentialHolders/info/{credentialHolderID}	Delete a credential holder
Credential holders	PUT	/AreaControl/CredentialHolders/info/{credentialHolderID}	Update a credential holder
Roles	GET	/AreaControl/Roles/info	Get role list
Roles	GET	/AreaControl/Roles/info/{roleUID}	Get a role
Roles	PUT	/AreaControl/Roles/info/{roleUID}	Update a role
Roles	DELETE	/AreaControl/Roles/info/{roleUID}	Delete a role
Credentials	GET	/AreaControl/Credentials/info	Get credential list
Credentials	POST	/AreaControl/Credentials/info	Create a credential
Credentials	GET	/AreaControl/Credentials/info/{credentialUID}	Get a credential
Credentials	PUT	/AreaControl/Credentials/info/{credentialUID}	Update a credential
Credentials	DELETE	/AreaControl/Credentials/info/{credentialUID}	Delete a credential
Doors	GET	/AreaControl/PartitionMembers/Portals/info	Get door list
Holiday	GET	/AreaControl/Holidays/info	Get Holiday List

Module	Method	Endpoint	Operation
Holiday	POST	/AreaControl/Holidays/info	Create a Holiday
Holiday	GET	/AreaControl/Holidays/info/{HolidayID}	Get a Holiday
Holiday	PUT	/AreaControl/Holidays/info/{HolidayID}	Update a Holiday
Holiday	DELETE	/AreaControl/Holidays/info/{HolidayID}	Delete a Holiday
TimeSchedule	GET	/AreaControl/TimeSchedules/info	Get TimeSchedule List
TimeSchedule	POST	/AreaControl/TimeSchedules/info	Create a TimeSchedule
TimeSchedule	GET	/AreaControl/TimeSchedules/info/{TimeScheduleID}	Get a TimeSchedule
TimeSchedule	PUT	/AreaControl/TimeSchedules/info/{TimeScheduleID}	Update a TimeSchedule
TimeSchedule	DELETE	/AreaControl/TimeSchedules/info/{TimeScheduleID}	Delete a TimeSchedule
Meta(Event sessions)	GET	/Metadata/sessionSupport	Get device node session parameters
Meta(Event sessions)	GET	/Metadata/stream	Create a default synchronization session
Meta(Event sessions)	POST	/Metadata/stream	Create an event session
Meta(Event sessions)	DELETE	/Metadata/stream	Delete all event sessions
Meta(Event sessions)	GET	/Metadata/stream/{streamID}	Get an event session (start session if synchronous, return parameters if asynchronous)
Meta(Event sessions)	DELETE	/Metadata/stream/{streamID}	Delete an event session

Appendix B. XML Request Examples

The following examples are representative XML request bodies available from the Swagger specification. Replace IDs, GUIDs, credentials, dates, times, and names with project-specific values.

POST /System/time/ntpServers - Create a NTP server

```
<?xml version="1.0" encoding="UTF-8"?>
<NTPServer xmlns="urn:psialliance-org" version="1.0">
  <id>5</id>
  <addressingFormatType>ipaddress</addressingFormatType>
  <ipAddress>120.25.115.20</ipAddress>
</NTPServer>
```

PUT /System/time/ntpServers/{id} - Update a NTP server

```
<?xml version="1.0" encoding="UTF-8"?>
<NTPServer xmlns="urn:psialliance-org" version="1.0">
  <id>5</id>
  <addressingFormatType>ipaddress</addressingFormatType>
  <ipv6Address>2001:da8:9000::81</ipv6Address>
</NTPServer>
```

POST /CSEC/AAA/users - Create an authorized user

```
<?xml version="1.0" encoding="UTF-8"?>
<CSECUser xmlns="urn:psialliance-org" version="1.0">
    <id>3</id>
    <userEnabled>true</userEnabled>
    <userGUID>{E6152A7C-41F8-8006-2728-0A1FB495FF46}</userGUID>
    <UserLogin>
        <username>admin</username>
        <password>21232f297a57a5a743894a0e4a801fc3</password>
    </UserLogin>
</CSECUser>
```

PUT /CSEC/AAA/users - Update authorized user list

```
<?xml version="1.0" encoding="UTF-8"?>
<CSECUserList xmlns="urn:psialliance-org" version="1.0">
  <CSECUser version="1.0">
    <id>1</id>
    <userEnabled>true</userEnabled>
    <userGUID>{ACB2DF36-2437-17EA-05BB-A15820E5C017}</userGUID>
    <UserLogin>
      <username>armatura</username>
      <password>e64b78fc3bc91bcbc7dc232ba8ec59e0</password>
    </UserLogin>
  </CSECUser>
  <CSECUser version="1.0">
    <id>2</id>
    <userEnabled>true</userEnabled>
    <userGUID>{A70E2268-8828-C957-C838-5831BEBDA584}</userGUID>
    <UserLogin>
      <username>installer</username>
```



```
      <password>e64b78fc3bc91bcbc7dc232ba8ec59e0</password>
    </UserLogin>
  </CSECUser>
  <CSECUser version="1.0">
    <id>3</id>
    <userEnabled>true</userEnabled>
    <userGUID>{E6152A7C-41F8-8006-2728-0A1FB495FF46}</userGUID>
    <UserLogin>
      <username>admin</username>
      <password>21232f297a57a5a743894a0e4a801fc3</password>
    </UserLogin>
  </CSECUser>
</CSECUserList>
```

PUT /CSEC/AAA/users/{id} - Update an authorized user (by id)

```
<?xml version="1.0" encoding="UTF-8"?>
<CSECUser xmlns="urn:psialliance-org" version="1.0">
    <id>3</id>
    <userEnabled>false</userEnabled>
    <userGUID>{E6152A7C-41F8-8006-2728-0A1FB495FF46}</userGUID>
    <UserLogin>
        <username>admin</username>
        <password>21232f297a57a5a743894a0e4a801fc3</password>
    </UserLogin>
</CSECUser>
```

PUT /CSEC/AAA/users/{GUID} - Update an authorized user (by GUID)

```
<?xml version="1.0" encoding="UTF-8"?>
<CSECUser xmlns="urn:psialliance-org" version="1.0">
    <id>3</id>
    <userEnabled>true</userEnabled>
    <userGUID>{E6152A7C-41F8-8006-2728-0A1FB495FF46}</userGUID>
    <UserLogin>
        <username>admin</username>
        <password>21232f297a57a5a743894a0e4a801fc3</password>
    </UserLogin>
</CSECUser>
```

POST /AreaControl/CredentialFormats/info - Create a Wiegand format

```
<?xml version="1.0" encoding="UTF-8"?>
<WiegandCardFormatInfo xmlns="urn:psialliance-org" version="1.0">
    <ID>5</ID>
    <UID>{F3F02378-8B2B-4416-BAD9-C972D89D6746}</UID>
    <Name>Wiegand Format26</Name>
    <Length>26</Length>
    <DataFieldList>
        <DataField>
            <Type>FacilityCode</Type>
            <Offset>1</Offset>
            <Length>8</Length>
            <Value>123</Value>
            <ValueEncoding>Decimal</ValueEncoding>
        </DataField>
        <DataField>
            <Type>CardNum</Type>
            <Offset>9</Offset>
            <Length>16</Length>
        </DataField>
        <ParityField>
            <ParityType>Even</ParityType>
            <CheckBitLocation>0</CheckBitLocation>
            <Start>0</Start>
            <Length>13</Length>
        </ParityField>
        <ParityField>
            <ParityType>Odd</ParityType>
            <CheckBitLocation>25</CheckBitLocation>
            <Start>13</Start>
            <Length>13</Length>
        </ParityField>
    </DataFieldList>
</WiegandCardFormatInfo>
```

PUT /AreaControl/CredentialFormats/info/{formatID} - Update a Wiegand format

```
<?xml version="1.0" encoding="UTF-8"?>
<WiegandCardFormatInfo xmlns="urn:psialliance-org" version="1.0">
    <ID>5</ID>
    <UID>{F3F02378-8B2B-4416-BAD9-C972D89D6746}</UID>
```



```
    <Name>Wiegand Format26</Name>
    <Length>26</Length>
    <DataFieldList>
        <DataField>
            <Type>FacilityCode</Type>
            <Offset>1</Offset>
            <Length>6</Length>
            <Value>12</Value>
            <ValueEncoding>Decimal</ValueEncoding>
        </DataField>
        <DataField>
            <Type>CardNum</Type>
            <Offset>7</Offset>
            <Length>18</Length>
        </DataField>
        <ParityField>
            <ParityType>Even</ParityType>
            <CheckBitLocation>0</CheckBitLocation>
            <Start>0</Start>
            <Length>13</Length>
        </ParityField>
        <ParityField>
            <ParityType>Odd</ParityType>
            <CheckBitLocation>25</CheckBitLocation>
            <Start>13</Start>
            <Length>13</Length>
        </ParityField>
    </DataFieldList>
</WiegandCardFormatInfo>
```

POST /AreaControl/Permissions/info - Create a credential permission

```
<?xml version="1.0" encoding="UTF-8"?>
<PermissionInfo xmlns="urn:psialliance-org" version="1.0">
  <ID>2</ID>
  <UID>{65DF0744-AE13-F22D-51C2-F44FD1D4AC55}</UID>
  <Name>test</Name>
  <PrivilegeList>
    <Privilege>
      <Allow>
        <TimeScheduleIDList>
          <TimeScheduleID>
            <ID>1</ID>
            <GUID>{55B0071D-4BAD-FE9A-A263-E39D5BAB9DC7}</GUID>
            <Name>24-Hour Accessible</Name>
          </TimeScheduleID>
        </TimeScheduleIDList>
        <PortalIDList>
          <PortalID>
            <ID>1</ID>
            <GUID>{D3B66429-9A99-49AD-8407-851F2B4C291A}</GUID>
            <Name>10.8.12.234-1</Name>
          </PortalID>
          <PortalID>
            <ID>2</ID>
            <GUID>{6A201845-F8A4-4905-9413-182022D48E34}</GUID>
            <Name>10.8.12.234-2</Name>
          </PortalID>
        </PortalIDList>
      </Allow>
    </Privilege>
  </PrivilegeList>
</PermissionInfo>
```

PUT /AreaControl/Permissions/info/{permissionID} - Update a credential permission

```
<?xml version="1.0" encoding="UTF-8"?>
<PermissionInfo xmlns="urn:psialliance-org" version="1.0">
  <ID>2</ID>
  <UID>{65DF0744-AE13-F22D-51C2-F44FD1D4AC55}</UID>
  <Name>test</Name>
  <PrivilegeList>
    <Privilege>
      <Allow>
        <TimeScheduleIDList>
          <TimeScheduleID>
            <ID>1</ID>
            <GUID>{55B0071D-4BAD-FE9A-A263-E39D5BAB9DC7}</GUID>
            <Name>24-Hour Accessible</Name>
          </TimeScheduleID>
```



```
        </TimeScheduleIDList>
        <PortalIDList>
          <PortalID>
            <ID>1</ID>
            <GUID>{D3B66429-9A99-49AD-8407-851F2B4C291A}</GUID>
            <Name>10.8.12.234-1</Name>
          </PortalID>
        </PortalIDList>
      </Allow>
    </Privilege>
  </PrivilegeList>
</PermissionInfo>
```

POST /AreaControl/CredentialHolders/info - Create a credential holder

```
<?xml version="1.0" encoding="UTF-8"?>
<CredentialHolderInfo version="1.0" xmlns="urn:psialliance-org">
  <ID>1</ID>
  <UID>{82E01D49-8D02-4FA5-8AB3-1302C8483DFE}</UID>
  <Name>Last456,FirstName456</Name>
  <GivenName>Last456</GivenName>
  <Surname>FirstName456</Surname>
  <Email>Last456@mail.com</Email>
  <CreationDate>2024-05-14T13:48:56</CreationDate>
  <ActiveFrom>2024-05-30T10:55:00</ActiveFrom>
  <ActiveTill>2024-12-30T10:55:00</ActiveTill>
  <State>Active</State>
  <RoleIDList>
    <RoleID>
      <ID>1</ID>
      <GUID>{A7F15B05-A443-F1EB-76B0-509663334711}</GUID>
      <Name>System Administration</Name>
    </RoleID>
  </RoleIDList>
</CredentialHolderInfo>
```

PUT /AreaControl/CredentialHolders/info/{credentialHolderID} - Update a credential holder

```
<?xml version="1.0" encoding="UTF-8"?>
<CredentialHolderInfo version="1.0" xmlns="urn:psialliance-org">
  <ID>1</ID>
  <UID>{82E01D49-8D02-4FA5-8AB3-1282C8483DDE}</UID>
  <Name>Last123,FirstName123</Name>
  <GivenName>Last123</GivenName>
  <Surname>FirstName123</Surname>
  <Email>Last123@mail.com</Email>
  <CreationDate>2024-01-14T14:48:56</CreationDate>
  <ActiveFrom>2024-01-01T10:55:00</ActiveFrom>
  <ActiveTill>2024-12-30T10:55:00</ActiveTill>
  <State>Active</State>
  <RoleIDList>
    <RoleID>
      <ID>2</ID>
      <GUID>{45297D5F-3DDF-1A28-B6B8-2BEDC7486A3C}</GUID>
      <Name>role2</Name>
    </RoleID>
  </RoleIDList>
</CredentialHolderInfo>
```

PUT /AreaControl/Roles/info/{roleUID} - Update a role

```
<?xml version="1.0" encoding="UTF-8"?>
<RoleInfo xmlns="urn:psialliance-org" version="1.0">
    <ID>2</ID>
    <UID>{45297D5F-3DDF-1A28-B6B8-2BEDC7486A3C}</UID>
    <Name>role2</Name>
    <PermissionIDList>
      <PermissionID>
        <ID>1</ID>
        <GUID>{297C3B46-E220-CE27-78DC-A42E01138547}</GUID>
        <Name>24-Hour Accessible</Name>
      </PermissionID>
      <PermissionID>
        <ID>2</ID>
        <GUID>{5747AA92-1193-D032-B9FD-AA78D62AA8E6}</GUID>
        <Name>8-Hour Accessible</Name>
      </PermissionID>
    </PermissionIDList>
  </RoleInfo>
```

POST /AreaControl/Credentials/info - Create a credential


```
<?xml version="1.0" encoding="UTF-8"?>
<CredentialInfo xmlns="urn:psialliance-org" version="1.0">
    <ID>1</ID>
    <UID>{abacab00-0000-1761-100a-761234500001}</UID>
    <Name>Credential 1</Name>
    <AssignedToID>
        <ID>1</ID>
        <GUID>{82E01D49-8D02-4FA5-8AB3-1282C8483DDE}</GUID>
    </AssignedToID>
    <State>Active</State>
    <LastModifiedDate>2024-10-10T11:05:10</LastModifiedDate>
    <IdentifierInfoList>
        <IdentifierInfo>
            <Type>Card</Type>
            <Value>1349537517</Value>
            <ValueEncoding>Decimal</ValueEncoding>
        </IdentifierInfo>
        <IdentifierInfo>
            <Type>PIN</Type>
            <Value>666666</Value>
        </IdentifierInfo>
    </IdentifierInfoList>
</CredentialInfo>
```

PUT /AreaControl/Credentials/info/{credentialUID} - Update a credential

```
<?xml version="1.0" encoding="UTF-8"?>
<CredentialInfo xmlns="urn:psialliance-org" version="1.0">
    <ID>1</ID>
    <UID>{abacab00-0000-1761-100a-761234500001}</UID>
    <Name>Credential 1</Name>
    <AssignedToID>
        <ID>1</ID>
        <GUID>{82E01D49-8D02-4FA5-8AB3-1282C8483DDE}</GUID>
    </AssignedToID>
    <State>Active</State>
    <LastModifiedDate>2024-10-10T11:05:10</LastModifiedDate>
    <IdentifierInfoList>
        <IdentifierInfo>
            <Type>Card</Type>
            <Value>10733853</Value>
            <ValueEncoding>Decimal</ValueEncoding>
        </IdentifierInfo>
        <IdentifierInfo>
            <Type>PIN</Type>
            <Value>123456</Value>
        </IdentifierInfo>
    </IdentifierInfoList>
</CredentialInfo>
```

POST /AreaControl/Holidays/info - Create a Holiday

```
<?xml version="1.0" encoding="UTF-8"?>
<HolidayInfo xmlns="urn:psialliance-org" version="1.0">
    <ID>7</ID>
    <UID>{62BC68F0-5F6F-49B0-ACC1-00001234500F}</UID>
    <Name>New Years Day</Name>
    <Description>January 1</Description>
    <RecursYearly>true</RecursYearly>
    <StartDate>2025-01-01</StartDate>
    <EndDate>2025-01-03</EndDate>
</HolidayInfo>
```

PUT /AreaControl/Holidays/info/{HolidayID} - Update a Holiday

```
<?xml version="1.0" encoding="UTF-8"?>
<HolidayInfo xmlns="urn:psialliance-org" version="1.0">
    <ID>7</ID>
    <UID>{2833AE4A-0BAF-CC6F-4E6C-30EE771519C5}</UID>
    <Name>Christmas Day</Name>
    <Description>December 25</Description>
    <RecursYearly>true</RecursYearly>
    <StartDate>2024-12-25</StartDate>
    <EndDate>2024-12-28</EndDate>
</HolidayInfo>
```

POST /AreaControl/TimeSchedules/info - Create a TimeSchedule

```
<?xml version="1.0" encoding="UTF-8"?>
<TimeScheduleInfo xmlns="urn:psialliance-org" version="1.0">
  <ID>2</ID>
  <UID>{76A628E1-EDF6-EAEC-7570-286E8B500D7E}</UID>
```



```
  <Name>8-Hour Accessible</Name>
  <Description>8-Hour Accessible</Description>
  <TimeIntervalInfoList>
    <TimeIntervalInfo>
      <Day>Monday</Day>
      <StartTime>09:00:00</StartTime>
      <EndTime>16:59:00</EndTime>
    </TimeIntervalInfo>
    <TimeIntervalInfo>
      <Day>Holiday</Day>
      <HolidayID>
        <ID>7</ID>
        <GUID>{62BC68F0-5F6F-49B0-ACC1-00001234500F}</GUID>
        <Name>New Years Day</Name>
      </HolidayID>
      <StartTime>10:00:00</StartTime>
      <EndTime>18:00:00</EndTime>
    </TimeIntervalInfo>
  </TimeIntervalInfoList>
</TimeScheduleInfo>
```

PUT /AreaControl/TimeSchedules/info/{TimeScheduleID} - Update a TimeSchedule

```
<?xml version="1.0" encoding="UTF-8"?>
<TimeScheduleInfo xmlns="urn:psialliance-org" version="1.0">
  <ID>2</ID>
  <UID>{76A628E1-EDF6-EAEC-7570-286E8B500D7E}</UID>
  <Name>8-Hour Accessible</Name>
  <Description>8-Hour Accessible</Description>
  <TimeIntervalInfoList>
    <TimeIntervalInfo>
      <Day>Holiday</Day>
      <HolidayID>
        <ID>7</ID>
        <GUID>{62BC68F0-5F6F-49B0-ACC1-00001234500F}</GUID>
        <Name>New Years Day</Name>
      </HolidayID>
      <StartTime>10:00:00</StartTime>
      <EndTime>18:00:00</EndTime>
    </TimeIntervalInfo>
  </TimeIntervalInfoList>
</TimeScheduleInfo>
```

POST /Metadata/stream - Create an event session

```
<?xml version="1.0" encoding="UTF-8"?>
<MetaSessionParms xmlns="urn:psialliance-org" version="1.1">
    <metaXportParms>
        <metaSessionID>0</metaSessionID>
        <metaFormat>xml-psia</metaFormat>
        <metaSessionType>RESTSyncSessionTargetSend</metaSessionType>
        <metaSessionFlowType>datastream</metaSessionFlowType>
    </metaXportParms>
</MetaSessionParms>
```


Appendix C. Terminology

Terms and Definitions	
ACS	Access Control System.
AHSC-1000 Controller	Armatura primary controller supported by the Phalanx Protocol SDK.
AHDU Series Controller	Armatura distributed controller series supported by the Phalanx Protocol SDK.
CSEC	PSIA Common Security specification used for device ownership and permission model.
Credential Holder	A person or identity record that can be assigned one or more credentials.
Credential	An access identifier such as card or PIN assigned to a credential holder.
Issuer-Signature	Security value used to validate authorized request issuer context.
PLAI	Physical Logical Access Interoperability, a PSIA specification for identity interoperability.
Portal/Door	A physical access point represented by a controller resource.
PSIA	Physical Security Interoperability Alliance.
RESTful API	HTTP-based API using standard methods such as GET, POST, PUT, and DELETE.
Swagger UI	Interactive OpenAPI documentation interface provided for API reference and testing.
Time Schedule	A defined set of time intervals used in access permissions.

