IBM

Spyre Support Appliance for IBM Z and LinuxONE User's Guide

GC28-7072-00

Note:

Before you use this information and the product it supports, read the information in “Safety” on page v, “Notices” on page 43, and *IBM Systems Environmental Notices and User Guide*, Z125–5823.

This edition, GC28-7072-00, applies to IBM z17 (Model ME1) and IBM LinuxONE Emperor 5 (Model ML1).

There might be a newer version of this document in a **PDF** file available on **IBM Documentation**. Go to https://www.ibm.com/docs/en/systems-hardware, select **IBM Z** or **IBM LinuxONE**, then select your configuration, and click **Library Overview** on the navigation bar.

© Copyright International Business Machines Corporation 2025.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Safety

Safety notices

Safety notices may be printed throughout this guide. **DANGER** notices warn you of conditions or procedures that can result in death or severe personal injury. **CAUTION** notices warn you of conditions or procedures that can cause personal injury that is neither lethal nor extremely hazardous. **Attention** notices warn you of conditions or procedures that can cause damage to machines, equipment, or programs.

World trade safety information

Several countries require the safety information contained in product publications to be provided in their local language(s). If this requirement applies to your country, a safety information booklet is included in the publications package shipped with the product. The booklet contains the translated safety information with references to the US English source. Before using a US English publication to install, operate, or service this product, you must first become familiar with the related safety information in the *Systems Safety Notices*, G229-9054. You should also refer to the booklet any time you do not clearly understand any safety information in the US English publications.

Laser safety information

All IBM Z® and IBM® LinuxONE (LinuxONE) models can use I/O cards such as FICON®, Open Systems Adapter (OSA), Network Express, Integrated Coupling Adapter2.0 SR (ICA SR2.0), zHyperLink Express, or other I/O features which are fiber optic based and utilize lasers (short wavelength or long wavelength lasers).

Laser compliance

All lasers are certified in the US to conform to the requirements of DHHS 21 CFR Subchapter J for Class 1 or Class 1M laser products. Outside the US, they are certified to be in compliance with IEC 60825 as a Class 1 or Class 1M laser product. Consult the label on each part for laser certification numbers and approval information.

Laser Notice: U.S. FDA CDRH NOTICE if low power lasers are utilized, integrated, or offered with end product systems as applicable. Complies with 21 CFR 1040.10 and 1040.11 except for conformance with IEC 60825-1 Ed. 3., as described in Laser Notice No. 56, dated May 8, 2019.

CAUTION: Data processing environments can contain equipment transmitting on system links with laser modules that operate at greater than Class 1 power levels. For this reason, never look into the end of an optical fiber cable or open receptacle. (C027)

CAUTION: This product contains a Class 1M laser. Do not view directly with optical instruments. (C028)

© Copyright IBM Corp. 2025

About this publication

This publication contains information about the IBM Spyre® Support Appliance on the 9175 system.

Unless otherwise stated, throughout this document "9175" refers to the IBM z17® (Model ME1) or IBM LinuxONE Emperor 5 (Model ML1).

Figures included in this document illustrate concepts and are not necessarily accurate in content, appearance, or specific behavior.

Related publications

Publications that you will find helpful and that you should use along with this publication are in the following list. The following publications are available on **IBM Documentation**. Go to https://www.ibm.com/docs/en/systems-hardware, select **IBM Z** or **IBM LinuxONE**, then select your configuration, and click **Library Overview** on the navigation bar.

- Appliance Control Center for IBM Z and LinuxONE User's Guide, GC28-7073
- Secure Service Container (SSC) User's Guide, SC28-7062
- IBM Dynamic Partition Manager (DPM) Guide

Related HMC and SE console information

Hardware Management Console (HMC) and Support Element (SE) information can be found on the console help system.

Accessibility features

Accessibility features help users who have physical disabilities such as restricted mobility or limited vision use software products successfully. The accessibility features can help users do the following tasks:

- Run assistive technology such as screen readers and screen magnifier software.
- Operate specific or equivalent features by using the keyboard.
- Customize display attributes such as color, contrast, and font size.

Consult assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in this product. Consult the product information for the specific assistive technology product that is used to access our product information.

Keyboard navigation

This product uses standard Microsoft Windows navigation keys.

IBM and accessibility

See http://www.ibm.com/able for more information about the commitment that IBM has to accessibility.

How to provide feedback to IBM

We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

For additional information use the following link that corresponds to your configuration:

© Copyright IBM Corp. 2025

Configuration	Link
IBM z17® Model ME1	https://www.ibm.com/docs/en/systems-hardware/zsystems/9175-ME1?topic=how-send-feedback
IBM LinuxONE Emperor 5 Model ML1	https://www.ibm.com/docs/en/systems-hardware/linuxone/9175-ML1?topic=how-send-feedback

Summary of changes

Summary of changes for the *Sypre Support Appliance for IBM Z and LinuxONE User's Guide* , GC28-7072.

Table 1. Summary of changes

Release level	Changes in level
November 2025	This revision contains editorial changes and the following technical changes: • "v1" has been added to the API endpoints. For more information, see “Certificate management” on page 36. • Use cases on configuring CA certificates have been added. For more information see, “Use Case 1 - Uploading a third-party signed server certificate” on page 14 and “Use Case 2- Obtaining SSA CA Signed by Third-Party CA and Generating a Server Certificate” on page 16

© Copyright IBM Corp. 2025

Chapter 1. Introduction to Spyre Support Appliance for IBM Z and IBM LinuxONE

The Spyre Support Appliance for IBM Z and IBM LinuxONE is a purpose-built solution for managing, monitoring, and diagnosing Spyre cards attached to an IBM Z Logical Partition (LPAR). The SSA continuously monitors all connected Spyre cards to maintain efficiency and enable proactive error handling.

When initialized, SSA activates all detected Spyre cards that are assigned to the LPAR. It monitors the system every second to detect newly added or removed cards and dynamically performs initialization or cleanup as needed. In addition to lifecycle management, SSA collects key metrics and health data for each Spyre card to maintain performance and detect issues early.

The SSA includes a comprehensive suite of REST APIs that enables administrators and tools to:

- Manage card states
- Retrieve error logs
- Run diagnostic routines and retrieve results
- Collect system-level and AIU-specific metrics

This guide outlines the core features of SSA, provides usage instructions for its REST interfaces, and explains how to interpret health and diagnostic information to ensure the effective management of Spyre hardware in your environment.

The topics that are covered in this section are:

System Requirements

This topic outlines the minimum hardware and software specifications that are required to install and run SSA, including storage, network, and compute requirements.

Storage:

Type: DASD or FCP

Minimum Boot Disk Size: 50 GB

Note: Dynamic storage pools are not supported by SSA.

Network:

Configuration: Defined in SE/HMC profile

Note: These settings cannot be changed at run time; it requires LPAR reactivation.

Supported Network devices: OSA (defined by PCHID) and NETH (defined by FID) devices

Compute:

CPU: 50% of the capacity of two shared IFLs during peak usage.

IP Configuration:

SSA supports IPv4 network configuration only.

Memory:

SSA requires a minimum of 50 GB of memory.

API rate limit:

SSA allows a maximum of 50 requests per second.

© Copyright IBM Corp. 2025

Securing Appliance

This section describes secure system practices, including multi-factor authentication (MFA), and details monitoring tools, log types, and health indicators for Spyre cards and services. An administrator can track authentication events, system changes, and reliability, availability, and serviceability (RAS) events by using the Appliance Control Center for IBM Z and IBM LinuxONE's (ACC) user interface. Best practices include enabling MFA, rotating secrets, and maintaining optimal disk usage.

Security Monitoring and Auditing

SSA monitoring and logging capabilities:

- Monitoring Tools: Event monitoring and secure logging by using SSA host
- Log Access: Use "System Logs" on SSA to access unencrypted system logs
- Types of Captured Logs:
  - Network activity logs
  - Storage logs
  - Web/Application server logs
  - User activity logs

Detailed log types:

- Card management logs: Tracks administrator activity that is related to the Spyre card management.
- Spyre card logs: Logged by Physical Function (PF) and Virtual Function (VF)
  - Format: /PF-ID/aiucardmgmt_<PF-ID>_PF.log
- Flight Logs: Generated for each Spyre card and stored in PF directory for system event tracing
- Metrics:
  - Aiumetrics: Detailed metrics data

Log Retention

- Logs are rotated by default every 90 days

Reliability, Availability, and Serviceability (RAS) events:

- Types: Hardware, software, configuration, Linux® config, unknown
- Visibility: Displayed in Appliance Control Center for IBM Z and IBM LinuxONE user interface under Spyre cards health monitor

Security Logging in SSA

Logged events:

- Authentication: Logins, logouts, failed logins, token failures
- System Changes: Resource modifications, service failures, reboots

SSA-specific logs:

- You can download the appliance logs from concurrent or disruptive dumps.
- You can download the Cardmgmt logs and Spyre card logs by using the APIs. For more information, see “Retrieving Logs” on page 27

RAS event logging:

- Hardware: Forward to System Engineer (SE) using SCLP ET24 and may trigger a call-home
- Software: Logged locally and retrievable using health monitoring APIs

Appliance Health Monitoring

Storage Health Monitor

- Monitoring interval is 5 minutes
- Status:
  - OK: <70%
  - DEGRADED: 70-90%
  - FAILED: >90%

Card Management Services

load-senlib-images (systemd service)

This service manages Senlib container images on the SSA. Restarting the service loads the latest Senlib images when a new z-images RPM is installed, helping ensure that the firmware deployed to Spyre cards is up to date.

Senlib Image:

A Senlib image is a containerized package that includes the firmware that is required to operate a Spyre card. It is essential for initializing and running the Spyre hardware.

aiucardmgmt (systemd service)

The aiucardmgmt service plays a central role in firmware deployment and Spyre card monitoring. It performs the following tasks:

- Periodically scans for Spyre cards that are attached to the logical partition (LPAR).
- Loads the appropriate firmware from the Senlib image onto the detected cards.
- Monitors the health of both Physical Functions (PFs) and Virtual Functions (VFs) of the Spyre cards.
- Detects and logs errors to help maintain card stability and performance.
- The monitoring interval is 5 seconds

Status definitions

- OK: Both services are active.
- DEGRADED: One service has failed.
- FAILED: Both services have failed, or the aiucardmgmt service has failed. Restart the appliance to restore functionality.

Spyre card Health Monitor

Each SSA logical partition (LPAR) supports up to 48 Spyre cards. After initialization by the SSA, the cards are ready to run workloads. During operation, the Spyre firmware continuously emits Reliability, Availability, and Serviceability (RAS) events. These events provide critical insights into the health and operational status of each Spyre card.

Note: It is recommended to have two SSAs sharing the Spyre cards for reliability, availability and serviceability.

The Spyre card Health Monitor component in the SSA continuously monitors all active Spyre cards in real time. It interprets Reliability, Availability, and Serviceability (RAS) events that are emitted by the firmware, identifies issues that may affect card stability or performance, and provides clear, actionable recommendations to help administrators resolve problems quickly.

Status indicators align with user interface (UI) color codes:

- OK - The Spyre card is healthy. No RAS events have occurred, and it is running workloads.
- Degraded - Warnings have been reported through RAS events. The card can still run workloads.

- Failed - Errors have been reported through RAS events. The card is not usable for workloads.

Spyre Card Status Indicators

ACC uses a color-coded status system in the UI to help administrators quickly assess the health of each Spyre card. These statuses are based on RAS (Reliability, Availability, and Serviceability) events detected by SSA.

Status	Description	Workload Capability
OK (Green)	The Spyre card is healthy. No RAS events have been detected.	✓ Workloads can run normally
Degraded (Yellow)	Warning-level RAS events have been observed. The card is still operational but may require attention.	✓ Workloads can run, but monitoring is advised
Failed (Red)	Error-level RAS events have occurred. The card is not functioning correctly.	× Card is not usable for workloads

CA-signed certificates

To secure communication and establish trust, SSA supports uploading a CA-signed certificate. Optionally, after installation, the SSA administrator can log in and upload the certificate using the certificate upload API. REST API clients must validate the certificate before accepting any data from SSA.

Best practices

This topic provides you with an overview of the best practices.

- Enable and regularly rotate multi-factor authentication (MFA) secrets.
- When an error occurs, analyze the logs to debug the issues.
- Check for Spyre card health status by using the ACC user interface regularly.
- Address card issues based on remediation suggestions.
- Restart the appliance only when necessary, such as in the event of service failures.
- Ensure that disk space is under 70% for optimal performance.

Chapter 2. Verifying Image Integrity

Spyre Support Appliance for IBM Z and IBM LinuxONE images are digitally signed to verify their authenticity and confirm their origin from IBM.

About this task

To confirm both the integrity and authenticity of the image, you must verify the digital signature.

Procedure

1. Log in to IBM Fix Central.
2. Download both the installation image and its corresponding signature file.
3. Download the public key from: https://public.dhe.ibm.com/systems/z/spyre_support_appliance/zssa-pub.pub
4. Run the following command to verify the image using OpenSSL:

   ```
   openssl dgst -sha256 -verify zssa-pub.pub -signature <signature_file> <image_file>
   ```

5. If the image is valid and untampered, you will see:

   ```
   Verified OK
   ```

   This confirms that the downloaded SSA image has not been altered and is indeed from IBM.

© Copyright IBM Corp. 2025

Chapter 3. Installing an Appliance using API commands

This section describes how to upload and install and appliance using Appliance Control Center for IBM Z and IBM LinuxONE.

Before you begin

- Verify the integrity of the image by following the instructions provided in Chapter 2, “Verifying Image Integrity,” on page 5.

Note: The process for installing appliances differs between the default and standalone modes of ACC. In default mode, no manual action is required on the HMC. In standalone mode, the LPARs must be set to active in "SSC Installer" mode through the HMC UI and must be accessible using the IP address configured in the LPAR’s network settings.

For a more detailed procedure, see the Secure Service Container (SSC) User's Guide available on IBM Documentation.

Procedure

1. Set Environment Variables

   Define the following environment variables on the control node:

Variable	Purpose
ACC_IP	IP address of the ACC
ACC_PORT	Port number (8081)
ACC_OWNER1_USERID	User ID of the appliance owner (assigned by ACC admin)
ACC_OWNER1_ASSIGNEDPASS	Initial password assigned by ACC admin
ACC_OWNER1_USERPASS	New password set by the owner
LPAR	Logical partition name for appliance installation
CORES	Number of cores (IFLs or GPs) for the LPAR
MEMORY	Amount of memory (in MiB) for the LPAR
APP_USERID	User ID to access the appliance
APP_USERPASS	Password to access the appliance
RP1	Name of the resource package that is assigned to the owner.
IMG_LOC	Location of the appliance image on control node

   You can store these in a shell script and run `source <script>` to load them into your environment.

2. Change Owner Password (First-Time Only)

   Use the password change API to update the assigned password:

```
curl -k -X 'PUT' "https://$ACC_IP:$ACC_PORT/api/user" \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "username": "'$ACC_OWNER1_USERID'",
    "old_password": "'$ACC_OWNER1_ASSIGNEDPASS'",
```

© Copyright IBM Corp. 2025

```
    "new_password": "'$ACC_OWNER1_USERPASS'"
  }'
```

Password must be 15–128 characters long, include uppercase, lowercase, digits, and special characters, and must not contain the username

3. Login to the ACC using owner credentials and obtain a token to perform authentication and authorization using `login API`.

```
response=$(curl -k -X 'POST' \
https://$ACC_IP:$ACC_PORT/api/user/token \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "username": "'$ACC_OWNER1_USERID'",
  "password": "'$ACC_OWNER1_USERPASS'"
}')
OWNER_TOKEN=$(echo $response | jq '.access_token' | tr -d '"')
```

The `$OWNER_TOKEN` will be used for authentication and authorization in subsequent steps.

4. Check for available resources that are assigned to the owner using [assigned resources API]()

```
curl -k -X GET \
    "https://$ACC_IP:$ACC_PORT/api/resource/pkgs" \
    -H "Authorization: Bearer $OWNER_TOKEN" \
    -H 'Content-Type: application/json'
```

The owner must ensure that the CPC and logical partitions intended for appliance installation are assigned to them by the ACC admin.

5. Upload appliance image

Upload the image to ACC and set its attributes.

For instance, the owner sets the `min_ifl` attribute to 1, which means that at least one IFL is required to activate the image.

```
curl -k -X 'POST' \
  "https://$ACC_IP:$ACC_PORT/api/images" \
  -H "Authorization: Bearer $OWNER_TOKEN" \
  -H "Content-Type: multipart/form-data" \
  -F "data=@${IMG_LOC}" \
  -F "image_type"="image" \
  -F "min_ifls=2" -F "min_memory=8192"
```

For new appliance installations, the `image_type` must be set to `'image'`. This indicates that the image located at `$IMG_LOC` will completely wipe the disk and install a fresh appliance.

.

6. You can retrieve image details from ACC using the Image Info API.

```
curl -k -X 'GET' \
  "https://$ACC_IP:$ACC_PORT/api/images" \
  -H "Authorization: Bearer $OWNER_TOKEN"
```

To display the output in a readable format, use the `jq` utility.

From the response, you can extract the `image_id` to specify which image to install.

Note: When replacing an existing image in ACC, a name or version conflict may occur. To resolve this, delete the existing image using the following API:

```
curl -k -X 'DELETE' \
"https://$ACC_IP:$ACC_PORT/api/images/<image_id>" \
-H "Authorization: Bearer $OWNER_TOKEN"
```

7. You can activate and install the appliance on an assigned LPARs.

```
curl -k --location "https://$ACC_IP:$ACC_PORT/api/cluster/activate" \
  --header "Content-Type: application/json" \
  --header "Authorization: Bearer $OWNER_TOKEN" \
```

```
  --data '{
    "$RP1": {
      "image_id": $IMG_ID,
      "processor_usage": "dedicated",
      "processor_type": "ifl",
      "memory": "$MEMORY",
      "cores": "$CORES",
      "username": "'$APP_USERID'",
      "password": "'$APP_USERPASS'",
      "hostname": "host1",
      "lpars": [
        {
          "name": "'$LPAR'",
          "execution_action": "default",
          "install": true
        }
      ]
    }
```

In this example, the appliance is installed using `image_id` 1 on the specified LPAR ($LPAR1), on resource package $RP1, with $CORES shared IFLs and $MEMORY MiB of memory.

The APP_USERID and APP_USERPASS fields represent the appliance management credentials

This command initiates the appliance installation process on ACC, which may take several minutes to complete.

You will get a task id. You can use the task id to verify the status of the appliance.

Results

Upon successful installation of SSA, all Spyre cards assigned to the SSA LPAR are automatically initialized using the default senlib (card firmware). This process is handled by the system, and no manual intervention is required from the administrator to load the senlib during a fresh installation.

Note: This automatic initialization ensures that the cards are ready for use immediately after installation, streamlining setup and reducing configuration time.

What to do next

Verify the appliance activation status by running the command:

```
curl -k -X GET \
    "https://$ACC_IP:$ACC_PORT/api/tasks/$TASK_ID/status" \
    -H "Authorization: Bearer $OWNER_TOKEN" | jq
```

If appliance activation is successful, then the status will change to "success". If activation fails, a rollback will be performed and lpar goes to "not activated" state.

Chapter 4. Managing Spyre Support Appliance for IBM Z and IBM LinuxONE

The topics in this section provide the instructions for managing the Spyre Support Appliance for IBM Z and IBM LinuxONE

Fetching Appliance Details

You can fetch appliance information by using the Spyre Support Appliance for IBM Z and IBM LinuxONE API.

Before you begin

Ensure that you have a valid API token and access to the Spyre Support Appliance for IBM Z and IBM LinuxONE API. For more information on generating an API token, see “API token” on page 31

Procedure

You can fetch the appliance details by running the following command:
Example request:

```
curl -k -X GET https://localhost/api/com.ibm.zaci.system/appliance \
 -H "Authorization: Bearer $token" \
 -H "zACI-API:com.ibm.zaci.system/1.0" \
 -H "Accept:application/vnd.ibm.zaci.payload+json;version=1.0"
```

Example response:

```
{
  "kind": "instance",
  "self": "/api/com.ibm.zaci.system/appliance",
  "resource-name": "appliance",
  "resource-version": "1.0",
  "properties": {
    "self": "/api/com.ibm.zaci.system/appliance",
    "name": "SSA",
    "description": "Spyre Support Appliance Image",
    "version": "1.1.2",
    "boot-time": 1757421507,
    "hostname": "hostname",
    "physical-server-name": "B89",
    "virtual-server-name": "SSACICD2",
    "vendor_expiration_date": "2026-09-07",
    "sdk_type": "zfab",
    "root-disk-info": {
      "device_type": "dasd",
      "device_id": "0.0.0f74"
    }
  }
}
```

Gathering concurrent dumps

You can gather concurrent dumps by using the curl commands.

Before you begin

Define the following variables:

© Copyright IBM Corp. 2025

Variable	Purpose
SSA_IP	IP address of SSA
SSA_LPAR_USERNAME	Username of the SSC LPAR on the HMC profile (used during SSA install)
SSA_LPAR_PASSWORD	Password of the SSC LPAR on the HMC profile (used during SSA install)
REASON	Reason for collecting SSA logs **Note:** This should not be more than 256 characters.

Procedure

1. Log in to the Appliance

   Use the SSC REST API to generate an authentication token:

   ```
   curl -k -X 'POST' \
     "https://$localhost/api/com.ibm.zaci.system/api-tokens" \
     -H 'Accept: application/vnd.ibm.zaci.payload+json' \
     -H 'Content-Type: application/vnd.ibm.zaci.payload+json;version=1.0' \
     -H 'zACI-API: com.ibm.zaci.system/1.0' \
     -d '{
           "user": "'$SSA_LPAR_USERNAME'",
           "password": "'$SSA_LPAR_PASSWORD'"
     }'
   ```

   If successful, store the returned token in a variable called `TOKEN`.

2. Send logging requests to SSA:

   Trigger log preparation by sending an alert:

   ```
   curl -k -X 'POST' \
     "https://localhost/api/com.ibm.zaci.system/alerts" \
     -H "Accept: application/vnd.ibm.zaci.payload+json" \
     -H "Content-Type: application/vnd.ibm.zaci.payload+json;version=1.0" \
     -H "zACI-API: com.ibm.zaci.system/1.0" \
     -H "Authorization: Bearer $TOKEN" \
     -d '{
           "reason": "'$REASON'",
           "diag_info": "concurrent"
     }'
   ```

   The response includes a `uuid`, which you should store in `UUID`.

3. Check Alert Status

   Monitor the alert status:

   ```
   curl -k -X 'GET' \
     "https://localhost/api/com.ibm.zaci.system/alerts/$UUID" \
     -H "Accept: application/vnd.ibm.zaci.payload+json" \
     -H "zACI-API: com.ibm.zaci.system/1.0" \
     -H "Authorization: Bearer $TOKEN"
   ```

   When the status is `200` or `202`, SSA has finished gathering logs.

4. Download the logs

   Download the log file:

   ```
   curl --fail-with-body -k -X 'GET' \
     "https://localhost/api/com.ibm.zaci.system/alerts/$UUID/diag-info" \
     -H "Accept: application/vnd.ibm.zaci.payload+json" \
     -H "zACI-API: com.ibm.zaci.system/1.0" \
     -H "Authorization: Bearer $TOKEN"
     --output SSA.tar.gz
   ```

   The output will be a compressed file in gz format.

Using CA-Signed Certificates

You can create a new ca-signed certificate or use an existing certificate with SSA.

Ensure that you have a valid API token and access to the Spyre Support Appliance for IBM Z and IBM LinuxONE API. For more information on generating API token, see “API token” on page 31

Certificates

Web certificates are used to allow customers to manage OpenSSL X.509 certificates used by the Nginx server.

Resource properties

A `certificates` instance is represented by the following fields:

Name	Type	Description
id	String	Unique id of the certificate resource.
fingerprint	String	Digest of the certificate.
serial	String	Certificate serial number, in hexadecimal.
ca	boolean	Differentiates CA and Leaf certificate. True for CA and False for Leaf.
hostname	String	Target host name, corresponds to the Common Name (CN) of the certificate.
names	list	Host name followed by list of alternative names of the certificate.
issued-by	String	Identifies the entity that has signed and issued the certificate.
issued-to	String	Identifies the entity that requested the certificate.
not-after	Timestamp	Ending date of certificate validity.
not-before	Timestamp	Starting date of certificate validity.
state	Enum String	Current State of the certificate.
csr	String	Certificate Signing Request in Base64 encoded format.
crt	String	Certificate content in Base64 encoded format.
self	String/URI	The URI of this instance.

Dictionary of certificate signing request (CSR) input properties

Name	Type	Description
c	String	Country code
st	String	State code
o	String	Organization name
ou	String	Organizational unit
ca	boolean	Differentiates CA and Leaf CSR. True for CA and False for Leaf.
ip	String	IP address

Enum of certificate states

State	Description
csr	Certificate Signing Request (CSR) is placed. But Certificate (CRT) is not generated or uploaded yet.
crt	Certificate (CRT) exists and is currently not active.
active	Certificate (CRT) exists and is active. Nginx is using this certificate.
expired	Certificate (CRT) is expired and unused.

SSA supports the following use cases:

- “Use Case 1 - Uploading a third-party signed server certificate” on page 14
- “Use Case 2- Obtaining SSA CA Signed by Third-Party CA and Generating a Server Certificate” on page 16

Use Case 1 - Uploading a third-party signed server certificate

This use case involves uploading a third-party signed server certificate to SSA for secure communication. First, generate a CSR on SSA with `ca: false` and include the IP address, then have it signed by the external CA. Concatenate the signed server certificate, intermediate certificate, and root certificate into a single PEM file and upload it. Finally, activate the new server certificate to establish a trusted connection.

Before you begin

- Ensure that the third-party root CA is present in the client truststore to validate the server certificate and establish a secure connection.
- Only PEM format (`text/plain`) is supported for certificate uploads. You can verify the file format by running the command:

```
% file ca.crt
ca.crt: PEM certificate
```

Procedure

1. Generate a CSR on the SSA,

   If `ca: false`, the CSR is for a server certificate, and you must provide the IP address.

   ```
   curl -k -X POST https://localhost/api/com.ibm.zaci.system/certificates/v1 \
   -H "Authorization: Bearer $token" \
   -H "Content-type: application/vnd.ibm.zaci.payload+json;version=1.0" \
   -H "zACI-API: com.ibm.zaci.system/1.0" \
   -H "Accept: application/vnd.ibm.zaci.payload+json;version=1.0" \
   -d '{"kind": "request","parameters": {"ca": false,"ip": "localhost","c": "IN","ou":
   "ibm.com","o": "ibm.com","st": "KA"}}' | jq
   ```

   You will receive a response similar to the following:

   ```
   {
     "kind": "instance",
     "self": "/api/com.ibm.zaci.system/certificates/v1",
     "resource-name": "certificates",
     "resource-version": "v1",
     "properties": {
       "issued-by": "C=IN, ST=KA, O=ibm.com, OU=ibm.com, CN=HOST-B02SSC2",
       "issued-to": "C=IN, ST=KA, O=ibm.com, OU=ibm.com, CN=HOST-B02SSC2",
       "state": "csr",
       "hostname": "N, ST=KA, O=ibm.com, OU=ibm.com, CN=HOST-B02SSC2",
       "ca": false,
       "names": [
         "N, ST=KA, O=ibm.com, OU=ibm.com, CN=HOST-B02SSC2",
         "HOST-B02SSC2",
         "localhost"
   ```

```
    ],
    "id": "d1a1837e-db04-4664-8976-1d20990497e6",
    "csr": "-----BEGIN CERTIFICATE REQUEST-----
     MIICyjCCAbICAQAwVTELMAkGA1UEBhMCSU4xCzAJBgNVBAgMAktBMRAwDgYDVQQKDAdpYm0uY29tMRAwDgYDV
     QQLDAdpYm0uY29tMRUwEwYDVQQDDAxIT1NULUIwMlNTQzIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
     AQDDDzP9nY57VshYWE7/6XdxnGhnIrdZEphUxz2uVzCEbvGbEXtA1teC5/BGKckZ0JvZeeHEl7+jUJktqI1af7oE

DMFFqsKSEGGY2lIw2MFZZvgUN2v9dRAA6HJkeOv8KLqF+VxZhmluMpOYj9u7hE4IVViX+5V+VCfxtuSJ19Xh13K/l
          -----END CERTIFICATE REQUEST-----"
  }
```

2. Sign the CSR.

   Copy the CSR and get it signed from 3rd party CA.

   Important:

   The CSR response is a single line. Add the newline after `-----BEGIN CERTIFICATE REQUEST-----` and before `-----END CERTIFICATE REQUEST-----`.

3. Concatenate Certificates.

   Combine the leaf/server certificate, intermediate certificate, and root certificate into a single file in PEM format.

```
curl -k -X PUT "https://localhost/api/com.ibm.zaci.system/certificates/v1
/d1a1837e-db04-4664-8976-1d20990497e6"
-H "Content-Type: text/plain; charset=utf-8"
-H "Authorization: Bearer $token" -H "zACI-API: com.ibm.zaci.system/1.0"
-H "Accept: application/vnd.ibm.zaci.payload+json;version=1.0"
--data-binary "@3rdpartysignedserver.pem" | jq
```

   The response will be similar to the following:

```
{
  "kind": "instance",
  "self": "/api/com.ibm.zaci.system/certificates/v1/d1a1837e-db04-4664-8976-1d20990497e6",
  "resource-name": "certificates",
  "resource-version": "v1",
  "properties": {
    "serial": "02FD57",
    "fingerprint": "E1:72:29:xx:49:xx:97:64:F3:8F:xx:4C:B8:1B:xx:09:9D:xx:58:FF",
    "issued-to": "C=IN, ST=KA, L=IBMBlr, O=ibm.com, OU=ibm.com, CN=HOST-B02SSC2,
UID=005O9P744, mail=emailid@ibm.com",
    "issued-by": "C=US, O=International Business Machines Corporation, CN=IBM INTERNAL
INTERMEDIATE CA",
    "not-before": 1762837200,
    "crt": "-----BEGIN CERTIFICATE-----
    +igAwIBAgIDAv1XMA0GCSqGSIb3DQEBCwUAMGoxCzAJBgNVBAYTAlVTMTQwMgYDVQQKEytJbnRlcm5hd
     hpbmVzIENvcnBvcmF0aW9uMSUwIwYDVQQDExxJQk0gSU5URVJOQUwgSU5URVJNRURJQVRFIENBMB4XD
     ExMTA0NTk1OVowgaYxCzAJBgNVBAYTAklOMQswCQYDVQQIEwJLQTEPMA0GA1UEBxMGSUJNQmxyMRAwD

     -----END CERTIFICATE-----",
    "not-after": 1825909199,
    "names": [
      "N, ST=KA, L=IBMBlr, O=ibm.com, OU=ibm.com, CN=HOST-B02SSC2, UID=005O9P744,
mail=email@ibm.com",
      "HOST-B02SSC2",
      "localhost"
    ],
    "state": "crt",
    "hostname": "N, ST=KA, L=IBMBlr, O=ibm.com, OU=ibm.com, CN=HOST-B02SSC2, UID=005O9P744,
mail=email@ibm.com",
    "ca": false,
    "id": "d1a1837e-db04-4664-8976-1d20990497e6",
    "csr": "-----BEGIN CERTIFICATEREQUEST-----
    MIICyjCCAbICAQAwVTELMAkGA1UEBhMCSU4xCzAJBgNVBAgMAktBMRAwDgYDVQQKDAdpYm0uY29tMRAwDgYD
    wYDVQQDDAxIT1NULUIwMlNTQzIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDDDzP9nY57VshYWE
    uVzCEbvGbEXtA1teC5/BGKckZ0JvZeeHEl7+jUJktqI1af7oEDMFFqsKSEGGY2lIw2MFZZvgUN2v9dRAA6HJ
    9u7hE4IVViX+5V+VCfxtuSJ19Xh13K/lh13dD8oFOjUas+jfc+vHInHO4K17IswvR7FAU5I7QNbGhKwMeN3+
    -----END CERTIFICATE REQUEST-----"
  }
```

   Tip: The id is used to activate the server certificates.

4. Activate the server certificate by using the id generated from “3” on page 15

```
curl -k -X POST "https://localhost/api/com.ibm.zaci.system/certificates/v1/
d1a1837e-db04-4664-8976-1d20990497e6?action=activate"
```

```
-H "Authorization: Bearer $token"
-H "zACI-API: com.ibm.zaci.system/1.0"
-H "Accept: application/vnd.ibm.zaci.payload+json;version=1.0"|jq
```

What to do next

Confirm that the new certificate chain is visible in the browser.

Use Case 2- Obtaining SSA CA Signed by Third-Party CA and Generating a Server Certificate

This use case covers obtaining an SSA CA certificate signed by a third-party CA and generating a server certificate. First, create a CA CSR on SSA and have it signed by the external CA, then upload the concatenated PEM file containing the CA, intermediate, and root certificates. Activate the new CA certificate, ensuring only one CA is active at a time. Finally, generate a CSR for the server certificate, sign it, and activate it for secure communication.

Before you begin

- Only PEM format (`text/plain`) is supported for certificate uploads.

Procedure

1. Generate a CSR on the SSA,

   If `ca: true`, the CSR is for a CA certificate and does not require an IP address.

   ```
   curl -k -X POST https://localhost/api/com.ibm.zaci.system/certificates/v1 \
   -H "Authorization: Bearer $token" \
   -H "Content-type: application/vnd.ibm.zaci.payload+json;version=1.0" \
   -H "zACI-API: com.ibm.zaci.system/1.0" \
   -H "Accept: application/vnd.ibm.zaci.payload+json;version=1.0" \
   -d '{"kind": "request","parameters": {"ca": true,"c": "IN","ou": "SSA","o": "IBM","st":
   "KA"}}' | jq
   ```

   You will receive a response similar to the following:

   ```
   {
     "kind": "instance",
     "self": "/api/com.ibm.zaci.system/certificates/v1",
     "resource-name": "certificates",
     "resource-version": "v1",
     "properties": {
       "issued-by": "C=IN, ST=KA, O=IBM, OU=SSA, CN=HOST-B02SSC2",
       "issued-to": "C=IN, ST=KA, O=IBM, OU=SSA, CN=HOST-B02SSC2",
       "state": "csr",
       "hostname": "N, ST=KA, O=IBM, OU=SSA, CN=HOST-B02SSC2",
       "ca": true,
       "names": [
         "N, ST=KA, O=IBM, OU=SSA, CN=HOST-B02SSC2",
         "HOST-B02SSC2"
       ],
       "id": "rootCA_da84d3b6-8474-4079-8775-98e95f961c99",
       "csr": "-----BEGIN CERTIFICATE REQUEST-----\nMIICvDCC
        AaQCAQAwTTELMAkGA1UEBhMCSU4xCzAJBgNVBAgMAktBMQwwCgYDVQQK\
        nDANJQk0xDDAKBgNVBAsMA1NTQTEVMBMGA1UEAwwMSE9TVC1CMDJTU0MyM
        IIBIjAN\nBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0ZmEOJNnavr+W
        UAH9VaQzGN5ges5\nCbqBScCdAGYEQQV19jtkgB6xMWTHa9ZYwpYffLA4b
        yqkO5lAnTUGNpXXpxL1QM=\n
             -----END CERTIFICATE REQUEST-----"
     }
   ```

2. Sign the CA CSR

   Ensure the third-party CA supports signing CA CSRs. If you do not have a third-party CA certificate, then you can generate a private certificate by following the instructions provided in “Generating a Private Self-Signed Root Certificate” on page 18

   Important:

The CSR response is a single line. Add the newline after `-----BEGIN CERTIFICATE REQUEST-----` and before `-----END CERTIFICATE REQUEST-----`.

3. Uploaded the concatenated certificates

   Concatenate the CA certificate, intermediate certificate, and root certificate into a single PEM file and upload.

```
curl -k -X PUT "https://localhost/api/com.ibm.zaci.system/certificates/v1/
rootCA_da84d3b6-8474-4079-8775-98e95f961c99"
-H "Content-Type: text/plain; charset=utf-8"
-H "Authorization: Bearer $token" -H "zACI-API: com.ibm.zaci.system/1.0"
-H "Accept: application/vnd.ibm.zaci.payload+json;version=1.0"
--data-binary "@rootCA_da84d3b6-8474-4079-8775-98e95f961c99.pem" | jq
```

   You will get a response similar to the following:

```
{
  "kind": "instance",
  "self": "/api/com.ibm.zaci.system/certificates/v1/
rootCA_da84d3b6-8474-4079-8775-98e95f961c99",
  "resource-name": "certificates",
  "resource-version": "v1",
  "properties": {
    "serial": "36C95CFC4EFEC05992B5E7CBD8C19564A8396560",
    "fingerprint": "BA:E0:72:2A:0E:45:AA:9E:99:FA:22:DD:99:F3:00:9B:65:A2:19:7A",
    "issued-to": "C=IN, ST=KA, O=IBM, OU=SSA, CN=HOST-B02SSC2",
    "issued-by": "C=IN, ST=KA, O=IBM, OU=SSA, CN=ItiRootCA",
    "not-before": 1762417087,
    "crt": "-----BEGIN CERTIFICATE-----
    \nMIIEZzCCAk+gAwIBAgIUNslc/E7+wFmStefL2MGVZKg5ZWAwDQYJKoZIhvcNAQEL
     \nBQAwSjELMAkGA1UEBhMCSU4xCzAJBgNVBAgMAktBMQwwCgYDVQQKDANJQk0xDDAK
     \nBgNVBAsMA1NTQTESMBAGA1UEAwwJSXRpUm9vdENBMB4XDTI1MTEwNjA4MTgwN1oX
     -----END CERTIFICATE-----",
    "not-after": 1793953087,
    "names": [
      "N, ST=KA, O=IBM, OU=SSA, CN=HOST-B02SSC2",
      ""
    ],
    "state": "crt",
    "hostname": "N, ST=KA, O=IBM, OU=SSA, CN=HOST-B02SSC2",
    "ca": true,
    "id": "rootCA_da84d3b6-8474-4079-8775-98e95f961c99",
    "csr": "-----BEGIN CERTIFICATE REQUEST-----
     \nMIICvDCCAaQCAQAwTTELMAkGA1UEBhMCSU4xCzAJBgNVBAgMAk
      tBMQwwCgYDVQQK\nDANJQk0xDDAKBgNVBAsMA1NTQTEVMBMGA1U
      EAwwMSE9TVC1CMDJTU0MyMIIBIjAN\nBgkqhkiG9w0BAQEFAAOC
             -----END CERTIFICATE REQUEST-----"
  }
```

Note:

4. Activate the new CA certificate.

   Only one CA certificate can be active at a time. Activating a new CA certificate automatically deactivates the previous one

```
curl -k -X POST "https://localhost/api/com.ibm.zaci.system/certificates/v1/
rootCA_da84d3b6-8474-4079-8775-98e95f961c99?action=activate"
-H "Authorization: Bearer $token"
-H "zACI-API: com.ibm.zaci.system/1.0"
-H "Accept: application/vnd.ibm.zaci.payload+json;version=1.0" | jq
```

5. Verify CA certificate activation.

   Check that `state=active` in the response.

```
curl -k -X GET https://localhost/api/com.ibm.zaci.system/certificates/v1/
rootCA_da84d3b6-8474-4079-8775-98e95f961c99
-H "Authorization: Bearer $token"
-H "zACI-API: com.ibm.zaci.system/1.0"
-H "Accept: application/vnd.ibm.zaci.payload+json;version=1.0" | jq
```

   The intermediate CA will be available for signing the server certificate.

6. Generate a CSR for the Server Certificate

```
curl -k -X POST https://localhost/api/com.ibm.zaci.system/certificates/v1 \
-H "Authorization: Bearer $token" \
-H "Content-type: application/vnd.ibm.zaci.payload+json;version=1.0" \
-H "zACI-API: com.ibm.zaci.system/1.0" \
-H "Accept: application/vnd.ibm.zaci.payload+json;version=1.0" \
-d '{"kind": "request","parameters": {"ca": false,"ip": "localhost","c": "IN",
"ou": "BN","o": "Test","st": "KA"}}' | jq
```

You will get a response similar to the following:

```
{
  "kind": "instance",
  "self": "/api/com.ibm.zaci.system/certificates/v1",
  "resource-name": "certificates",
  "resource-version": "v1",
  "properties": {
    "issued-by": "C=IN, ST=KA, O=IBM, OU=SSA, CN=HOST-B02SSC2",
    "issued-to": "C=IN, ST=KA, O=IBM, OU=SSA, CN=HOST-B02SSC2",
    "state": "csr",
    "hostname": "N, ST=KA, O=IBM, OU=SSA, CN=HOST-B02SSC2",
    "ca": false,
    "names": [
      "N, ST=KA, O=IBM, OU=SSA, CN=HOST-B02SSC2",
      "HOST-B02SSC2",
      "localhost"
    ],
    "id": "771ac4d4-aff7-4448-af53-25ee1e56e4c6",
    "csr": "-----BEGIN CERTIFICATE REQUEST-----
     \nMIICwjCCAaoCAQAwTTELMAkGA1UEBhMCSU4xCzAJBgNVBAgMAktBMQwwCgYDVQQK
     \nDANJQk0xDDAKBgNVBAsMA1NTQTEVMBMGA1UEAwwMSE9TVC1CMDJTU0MyMIIBIjAN\
     -----END CERTIFICATE REQUEST-----"
  }
```

7. Self sign the certificate by using the id from “6” on page 17.

```
curl -k -X POST "https://localhost/api/com.ibm.zaci.system/certificates/v1/
771ac4d4-aff7-4448-af53-25ee1e56e4c6?action=selfsign"
-H "Authorization: Bearer $token"
-H "zACI-API: com.ibm.zaci.system/1.0"
-H "Accept: application/vnd.ibm.zaci.payload+json;version=1.0"|jq
```

8. Activate the server certificate by using the id from “6” on page 17.

   Only one server certificate can be active at a time. Activating a new certificate automatically deactivates the previous one.

```
curl -k -X POST "https://localhoss/api/com.ibm.zaci.system/certificates/v1/
771ac4d4-aff7-4448-af53-25ee1e56e4c6?action=activate"
-H "Authorization: Bearer $token"
-H "zACI-API: com.ibm.zaci.system/1.0"
-H "Accept: application/vnd.ibm.zaci.payload+json;version=1.0"|jq
```

Generating a Private Self-Signed Root Certificate

You can generate a self-signed certificate on Linux or MAC Os by following the instructions provided in this topic.

Procedure

1. Generate RSA key pair by running the command:

```
openssl genrsa -out ca.key 4096
```

2. Generate a self-signed root certificate by running the command:

```
openssl req -x509 -new -nodes -sha256 -days 3650 -key ca.key -out ca.crt
-subj "/C=US/ST=State/L=City/O=ExampleOrg/OU=RootCA/CN=Example Root CA"
```

3. Create an intermediate CA extension configuration.

   This file is used to provide signing permissions when signing the CA CSR.

```
printf "[ext]\nbasicConstraints=critical,CA:TRUE,
pathlen:0\nkeyUsage=critical, digitalSignature, cRLSign,
keyCertSign\nsubjectKeyIdentifier=hash\nauthorityKeyIdentifier=keyid,issuer\n" > intCA.cfg
```

4. Sign the CA CSR.

```
openssl x509 -req -in intCA.csr -CA ca.crt -CAkey ca.key -CAcreateserial
-out intCA.crt -days 1825 -sha256 -extfile intCA.cfg -extensions ext
```

5. Concatenate the root ca certificate to the generated `intCA.crt`

```
cat ca.crt >> intCA.crt
```

What to do next

Use the `intCA.crt` file to generate a server certificate. For more information see, “Use Case 2- Obtaining SSA CA Signed by Third-Party CA and Generating a Server Certificate” on page 16

Chapter 5. Managing Spyre Cards using SSA

The topics in this section provide the instructions for managing the Spyre cards using Spyre Support Appliance for IBM Z and IBM LinuxONE.

Activating Spyre Card

You can activate one or all Spyre cards by using the Spyre Support Appliance for IBM Z and IBM LinuxONE API.

Before you begin

Ensure that you have a valid API token and access to the Spyre Support Appliance for IBM Z and IBM LinuxONE API. For more information on generating an API token, see "API token" on page 31

Procedure

1. List available Spyre Cards

   Use the GET `/spyre-cards` API to retrieve all available Spyre card IDs.

   Example request:

   ```
   GET /api/com.ibm.zaci.system/spyre-cards HTTPS/1.1
   Host: localhost:443
   Authorization: Bearer **********
   ZACI-API: com.ibm.zaci.system/1.0
   Accept: application/vnd.ibm.zaci.payload+json;version=1.0
   ```

   Example response:

   ```
   HTTPS/1.1 200 OK
   Date: XXX, XX XXX XXXX XX:XX:XX GMT
   ZACI-API: com.ibm.zaci.system/1.0
   Content-type: application/vnd.ibm.zaci.payload+json;version=1.0
   Content-length: 68
   {
     "kind": "response",
     "parameters": [
       "0000:00:00.0",
       "0001:00:00.0"
     ]
   }
   ```

2. Select the activation type

Activation type	API endpoint

Activate a specific card.	`POST /spyre-cards/activate?spyre_card_id=0000:00:00.0`
	Example request
	`curl -k -X POST https://9x.xx.xx.1x2/api/com.ibm.zaci.system /spyre-cards/activate?spyre_card_id=0000:00:00.0 \`

© Copyright IBM Corp. 2025

Activation type	API endpoint

	`-H "Authorization: Bearer $token" \`
	`-H "zACI-API: com.ibm.zaci.system/1.0" \`
	`-H "Accept:application/vnd.ibm.zaci.payload+json;version=1.0"`

All cards	`POST /spyre-cards/activate?spyre_card_id=all`
	Example request
	`curl -X POST "https://localhost/api/com.ibm.zaci.system/spyre-cards/activate?spyre_card_id=all" \`
	`-H "zACI-API: com.ibm.zaci.system/1.0" \`
	`-H "Authorization: Bearer $token>"`

202: Accepted.

Indicates that the activation request was successfully accepted.

3. Get the card details to check the card activation or deactivation status by running `GET /spyre-cards`

```
curl -k -X GET "https://localhost/api/com.ibm.zaci.system/
spyre-cards?spyre_card_id=all" \
-H "Authorization: Bearer $token" \
-H "zACI-API:com.ibm.zaci.system/1.0" \
-H "Accept:application/vnd.ibm.zaci.payload+json;version=1.0"
```

Deactivating Spyre Card

You can deactivate one or all Spyre cards by using the Spyre Support Appliance for IBM Z and IBM LinuxONE API.

Before you begin

Ensure you have a valid API token and access to the IBM Spyre Cards API. For more information on generating API token, see “API token” on page 31

Procedure

1. List available Spyre Cards

   Use the `GET /spyre-cards` API to retrieve all available Spyre card IDs.

   Example request:

```
GET /api/com.ibm.zaci.system/spyre-cards HTTPS/1.1
Host: localhost:443
Authorization: Bearer **********
ZACI-API: com.ibm.zaci.system/1.0
Accept: application/vnd.ibm.zaci.payload+json;version=1.0
```

   Example response:

```
HTTPS/1.1 200 OK
Date: XXX, XX XXX XXXX XX:XX:XX GMT
ZACI-API: com.ibm.zaci.system/1.0
Content-type: application/vnd.ibm.zaci.payload+json;version=1.0
Content-length: 68
{
  "kind": "response",
  "parameters": [
    "0000:00:00.0",
    "0001:00:00.0"
  ]
}
```

2. Select the deactivation type

Deactivation type	API endpoint

Deactivate a specific card.	`POST /spyre-cards/deactivate?spyre_card_id=<spyre_card_id>`
	Example request
	`curl -k -X POST https://localhost/api/com.ibm.zaci.system/spyre-cards/deactivate?spyre_card_id=0000:00:00.0 -H "Authorization: Bearer $token" -H "zACI-API: com.ibm.zaci.system/1.0" -H "Accept:application/vnd.ibm.zaci.payload+json;version=1.0"`

All cards	`POST /spyre-cards/deactivate?spyre_card_id=all`
	Example request
	`curl -X POST "https://localhost/api/com.ibm.zaci.system/spyre-cards/deactivate?spyre_card_id=all" \ -H "zACI-API: com.ibm.zaci.system/1.0" \ -H "Authorization: Bearer <your_token>"`

202: Accepted.

Indicates that the deactivation request was successfully accepted.

Updating Spyre Card Configuration

This task describes how to update the configuration of Spyre cards by selecting and applying available Senlib/libVF versions (card firmware) by using the Spyre Support Appliance for IBM Z and IBM LinuxONE API. Updating the Spyre Card configuration is a global operation. Once applied, the new configuration becomes available across the entire SSA environment for all Spyre Cards. To apply the changes, each Spyre Card must be deactivated and then reactivated.

Before you begin

Ensure that you have a valid API token and access to the Spyre Support Appliance for IBM Z and IBM LinuxONE API. For more information on generating API token, see “API token” on page 31

About this task

- The updated configuration does not automatically apply to active Spyre Cards. Cards that are currently active continue to run with their existing firmware until they are restarted. To apply the updated configuration, the administrator must deactivate and then reactivate each Spyre Card. This process reloads the firmware based on the new configuration.

Procedure

1. Get the available senlib/libvf versions by using the following API call:

```
curl -k -X GET "https://localhost/api/com.ibm.zaci.system/spyre-cards/senlib-images"
-H "Authorization: Bearer $token"
-H "zACI-API:com.ibm.zaci.system/1.0"
-H "Accept:application/vnd.ibm.zaci.payload+json;version=1.0"
```

Example request:

```
GET /api/com.ibm.zaci.system/spyre-cards/senlib-images HTTPS/1.1
Host: localhost:443
Authorization: Bearer **********
```

```
zACI-API: com.ibm.zaci.system/1.0
Accept: application/vnd.ibm.zaci.payload+json;version=1.0
```

Example response:

```
HTTPS/1.1 200 OK
Date: XXX, XX XXX XXXX XX:XX:XX GMT
zACI-API: com.ibm.zaci.system/1.0
Content-Type: application/vnd.ibm.zaci.payload+json;version=1.0
Content-length:195
{
  "kind": "response",
  "parameters": [
    "localhost/spyredriver:0.0.245.3f828e75a9e550xxc09cxdb137fd6cxxxaca4970"
  ]
}
```

2. Update the configuration by using the following API call:

```
PUT /spyre-cards/update-config
curl -X PUT "https://localhost/api/com.ibm.zaci.system/spyre-cards/update-config" \
  -H "zACI-API: com.ibm.zaci.system/1.0" \
  -H "Authorization: Bearer <your_token>" \
  -H "Content-Type: application/json" \
  -d '{{
    "spyre_card_id": "card01",
    "senlib_image": "senlib-v2.3.1",
    "libvf_image": "libvf-v1.4.0"
  }}'
```

3. Deactivate the card by following the instructions provided in “Deactivating Spyre Card” on page 22
4. Activate the card by following the instructions provided in “Activating Spyre Card” on page 21
5. Verify the version to list available Spyre Cards

   Use the `GET /spyre-cards?spyre_card_id=0000:00:00.0` API to retrieve all specific Spyre card IDs.

   Example request:

```
GET /api/com.ibm.zaci.system/spyre-cards?spyre_card_id=0000:00:00.0 HTTPS/1.1
Host: x.x.x.x:443
Authorization: Bearer **********
ZACI-API: com.ibm.zaci.system/1.0
Accept: application/vnd.ibm.zaci.payload+json;version=1.0
```

   Example response:

```
HTTPS/1.1 200 OK
Date: XXX, XX XXX XXXX XX:XX:XX GMT
ZACI-API: com.ibm.zaci.system/1.0
Content-type: application/vnd.ibm.zaci.payload+json;version=1.l
Content-length: 176
{
  "kind": "response",
  "parameters": {
    "0000:00:00.0": {
      "bus_id": "0000:00:00.0",
      "status": "online",
      "pftag": "localhost/spyredriver:0.0.245.3f828e75a9e550d6c09c8eb137fd6c337aca4970",
      "vftag": "localhost/spyredriver:0.0.245.3f828e75a9e550d6c09c8eb137fd6c337aca4970",
      "pfloglevel": "info",
      "vfloglevel": "info"
    }
  }
}
```

Running diagnostics

You can run diagnostics on a Spyre card that is in an offline state by using the Spyre Support Appliance for IBM Z and IBM LinuxONE API.

Before you begin

- Ensure you have a valid API token and access to the Spyre Support Appliance for IBM Z and IBM LinuxONE API. For more information on generating API token, see “API token” on page 31
- Ensure that the Spyre card is in offline state.

Procedure

1. List available Spyre Cards

   Use the GET `/spyre-cards` API to retrieve all available Spyre card IDs.

   Example request:

```
GET /api/com.ibm.zaci.system/spyre-cards HTTPS/1.1
Host: localhost:443
Authorization: Bearer **********
ZACI-API: com.ibm.zaci.system/1.0
Accept: application/vnd.ibm.zaci.payload+json;version=1.0
```

   Example response:

```
HTTPS/1.1 200 OK
Date: XXX, XX XXX XXXX XX:XX:XX GMT
ZACI-API: com.ibm.zaci.system/1.0
Content-type: application/vnd.ibm.zaci.payload+json;version=1.0
Content-length: 68
{
  "kind": "response",
  "parameters": [
    "0000:00:00.0",
    "0001:00:00.0"
  ]
}
```

2. Parse the response and identify cards with the status "offline".
3. Validate that the selected card is still offline by checking its status.
4. List available diagnostics by using the following API call:

```
curl -k -X GET https://localhost/api/com.ibm.zaci.system/spyre-cards/diag-utils
-H "Authorization: Bearer $token"
-H "zACI-API:com.ibm.zaci.system/1.0"
-H "Accept:application/vnd.ibm.zaci.payload+json;version=1.0"
```

   Example request:

```
GET /api/com.ibm.zaci.system/spyre-cards/diag-utils
HTTPS/1.1
Host: localhost:443
Authorization: Bearer **********
zACI-API: com.ibm.zaci.system/1.0
Accept: application/vnd.ibm.zaci.payload+json;version=1.0
```

   Example response:

```
HTTP/1.1 200 OK
zACI-API: com.ibm.zaci.system/1.0
Content-Type: application/vnd.ibm.zaci.payload+json;version=1.0
Content-Length: 1277
Date: XXX, XX XXX XXXX XX:XX:XX GMT
{
  "kind": "response",
  "parameters": {
    "senlib_ut_all_small": "/opt/sentient/senlib/bin/senlib_unit_test --gtest_filter='Sml*'",
    "senlib_ut_all_medium": "/opt/sentient/senlib/bin/senlib_unit_test --
```

```
gtest_filter='Med*'",
    "senlib_ut_all_large": "/opt/sentient/senlib/bin/senlib_unit_test --gtest_filter='Lrg*'",
    "senlib_ut": "/opt/sentient/senlib/bin/senlib_unit_test",
    "hot_reset": "/opt/sentient/senlib/bin/aiu_dd2_hot_reset -t chip",
    "connectivity": "/opt/sentient/senlib/bin/senlib_unit_test --
gtest_filter='SmlPF1VF0.OpenDiag'",
    "dma_tests": "/opt/sentient/senlib/bin/senlib_unit_test
--gtest_filter=*SmlPF1VF0.CmptEightRequests*:*SmlTest/SmlPF1VFx.ImmFillDmaIO/
0*:*SmlTest/SmlPF1VFx.ImmPostWipeDmaO/0*:*MedTest/MedPF1VFx.ImmDataDmaIO/0*:*LrgTest/
LrgPF1VFx.ManufMemDataDmaIO/0*",
    "rcu_tests": "/opt/sentient/senlib/bin/senlib_unit_test --gtest_filter=SmlTest/
SmlPF1VFx.ImmCmptBatchNorm/0",
    "sml_mem_tests": "/opt/sentient/senlib/bin/senlib_unit_test --gtest_filter=*SmlTest/
SmlPF1VFx.Mbist/0*:*SetTimeTest/SetTimePF1VF0.Vali* Connection #0 to
host <IP_Address> left intact dateLPDDR/2*",
    "med_mem_tests": "/opt/sentient/senlib/bin/senlib_unit_test --gtest_filter=SetTimeTest/
SetTimePF1VF0.ValidateLPDDR/6",
    "lrg_mem_tests": "/opt/sentient/senlib/bin/senlib_unit_test --gtest_filter=SetTimeTest/
SetTimePF1VF0.ValidateLPDDR/8"
  }
}
```

5. Start the diagnostics on a card

```
curl -k -X POST "https://localhost/api/com.ibm.zaci.system/spyre-cards/diag-utils?
spyre_card_id=0000:00:00.0" \
-H "Authorization: Bearer $token" \
-H "zACI-API: com.ibm.zaci.system/1.0" \
-H "Content-type: application/vnd.ibm.zaci.payload+json;version=1.0" \
-H "Accept: application/vnd.ibm.zaci.payload+json;version=1.0" \
-d '{"kind": "request", "parameters": {"diag-utils":"senlib_ut_all_small"}}'
```

Example request:

```
POST /api/com.ibm.zaci.system/spyre-cards/diag-utils?spyre_card_id=0000:00:00.0
HTTPS/1.1
Host: localhost:443
Authorization: Bearer **********
zACI-API: com.ibm.zaci.system/1.0
Content-type: application/vnd.ibm.zaci.payload+json;version=1.0
Accept: application/vnd.ibm.zaci.payload+json;version=1.0
Content-Length: 71
```

Example response:

```
HTTP/1.1 200 OK
zACI-API: com.ibm.zaci.system/1.0
Content-Type: application/vnd.ibm.zaci.payload+json;version=1.0
Content-Length: 122
Date: XXX, XX XXX XXXX XX:XX:XX GMT
{
  "kind": "response",
  "parameters": {
    "self": "/api/com.ibm.zaci.system/spyre-cards/diag-utils?spyre_card_id=0000:00:00.0"
  }
}
```

6. Check the card status periodically by using

```
GET /spyre-cards?spyre_card_id=0000:00:00.0
curl -X GET "https://localhost/api/com.ibm.zaci.system/spyre-cards?spyre_card_id=card01" \
  -H "zACI-API: com.ibm.zaci.system/1.0" \
  -H "Authorization: Bearer $token"
```

a) If diagnostics are running too long, terminate them

```
curl -k -X DELETE "https://localhost/api/com.ibm.zaci.system/
spyre-cards/diag-utils?spyre_card_id=0000:00:00.0"
-H "Authorization: Bearer $token"

-H "zACI-API: com.ibm.zaci.system/1.0"
-H "Accept:application/vnd.ibm.zaci.payload+json;version=1.0"
```

7. Retrieve diagnostic results by using the following API call:

```
curl -k -X GET https://localhost/api/com.ibm.zaci.system/
spyre-cards/diag-results?spyre_card_id=0000:00:00.0
-H "Authorization: Bearer $token"
-H "zACI-API: com.ibm.zaci.system/1.0"
-H "Accept:application/vnd.ibm.zaci.payload+json;version=1.0" --output /root/test_diag.gz"
```

Retrieving Logs

You can retrieve logs from Spyre card by using the Spyre Support Appliance for IBM Z and IBM LinuxONE API.

Before you begin

Ensure that you have a valid API token and access to the Spyre Support Appliance for IBM Z and IBM LinuxONE API. For more information on generating API token, see “API token” on page 31.

Procedure

1. List available Spyre Cards

   Use the `GET /spyre-cards` API to retrieve all available Spyre card IDs.

   Example request:

   ```
   GET /api/com.ibm.zaci.system/spyre-cards HTTPS/1.1
   Host: localhost:443
   Authorization: Bearer **********
   ZACI-API: com.ibm.zaci.system/1.0
   Accept: application/vnd.ibm.zaci.payload+json;version=1.0
   ```

   Example response:

   ```
   HTTPS/1.1 200 OK
   Date: XXX, XX XXX XXXX XX:XX:XX GMT
   ZACI-API: com.ibm.zaci.system/1.0
   Content-type: application/vnd.ibm.zaci.payload+json;version=1.0
   Content-length: 68
   {
     "kind": "response",
     "parameters": [
       "0000:00:00.0",
       "0001:00:00.0"
     ]
   }
   ```

2. Select the logs collection type

Logs type	API endpoint

Card management	`GET /spyre-cards/logs/card_mgmt`
	Example request
	`curl -X GET "https://localhost/api/com.ibm.zaci.system/ spyre-cards/logs/card_mgmt" \ -H "zACI-API: com.ibm.zaci.system/1.0" \ -H "Authorization: Bearer $token"`

card-logs	`GET /spyrecards/logs?spyre_card_id=0000:00:00.0`
	Example request
	`curl -k -X GET "https://localhost/api/com.ibm.zaci.system/ spyre-cards/logs?spyre_card_id=0000:00:00.0" -H "Authorization: Bearer $token" -H "zACI-API: com.ibm.zaci.system/1.0" -H "Accept:application/vnd.ibm.zaci.payload+json;version=1.0" --output /root/test_log.gz`

Logs type	API endpoint

All	`GET /spyre-cards/logs/all`
	Example request
	`curl -X GET "https://localhost/api/com.ibm.zaci.system/spyre-cards/logs/all" \ -H "zACI-API: com.ibm.zaci.system/1.0" \ -H "Authorization: Bearer $token"`

Download flight logs	`GET spyre-cards/logs/flight_logs?spyre_card_id=0000:00:00.0`
	Example request
	`curl -k -X GET "https://localhost/api/com.ibm.zaci.system/spyre-cards/logs/flight_logs?spyre_card_id=0001:00:00.0" -H "Authorization: Bearer $token" -H "zACI-API: com.ibm.zaci.system/1.0" -H "Accept: application/vnd.ibm.zaci.payload+json;version=1.0" --output flight-logs-0001-00-00-0.tar.gz`

Telemetry data	`GET spyre-cards/telemetry`
	Example request
	`curl -k -X GET https://localhost/api/com.ibm.zaci.system/spyre-cards/telemetry -H "Authorization: Bearer $token" -H "zACI-API: com.ibm.zaci.system/1.0" -H "Accept: application/vnd.ibm.zaci.payload+json;version=1.0" --output telemetry-data.tar.gz`

Note: SSA stores the last 10 executions of both diagnostic and flight logs.

Spyre Card Health Monitor

The Spyre Card Health Monitor continuously checks the operational status and health of all Spyre cards installed on the appliance. It ensures that each card is functioning correctly and provides visibility into issues such as offline cards, degraded performance, or hardware errors.

The health monitor performs the following steps automatically:

1. Verifies system readiness

   Ensures that the internal card management service (`aiucardmgmt`) is running.

   - If the service is down, the monitor enters a **Failed** state and displays a message prompting the user to check the service status.
2. Collects card information

   Retrieves a list of all Spyre cards and gathers status and identifiers for each card.
3. Evaluates card health

   For each card, the monitor:

   - Checks whether the card is online or offline.
   - Retrieves recent RAS (Reliability, Availability, and Serviceability) events and evaluates their severity (*Info*, *Degraded*, or *Error*).
   - Determines the card's health based on these events.
4. Reports results

   Summarizes the overall health of the cards using one of three monitor states.

State	Description	Recommended Action
OK	All Spyre cards are online and operating normally. No recent RAS events indicate any issues.	No action required. The system and all cards are functioning as expected.
Degraded	One or more Spyre cards are online but have recent RAS events with severity Warning or Info. These may indicate minor or early-stage issues.	Review the affected cards for additional details. Check logs or RAS events to understand the cause and take preventive actions if needed.
Failed	One or more Spyre cards are offline or have encountered critical errors (RAS events with severity Error).	Review the affected cards. For offline cards, refer to the action message, verify the card's status and connection, and check logs. If the issue persists, contact support.

Notes:

- An offline card is always reported as Failed, regardless of whether it was taken offline intentionally or due to a fault condition.
- The generic action message in the "Failed" state provides guidance to verify service status and review card logs.
- For advanced troubleshooting, use the `/ras` API or contact support with the card identifier and the timestamp of the last recorded event.

Chapter 6. Rest API Reference

This topics provides a list of API and their endpoints.

API token

The API token is used to authenticate with the SSA system and is required for all subsequent API operations.

DELETE /api-tokens

This API invalidates a valid API token.

Parameters
None

Responses:
204: Token is invalidated.

GET /api-tokens

This API validates an API token with a zFeature Appliance Base appliance.

Parameters
None

Responses:
204: The token is valid.

POST /api-tokens

This API authenticates with the appliance and returns an API tokens for subsequent requests. The token is required for subsequent API operations that require authentication.

Tokens generated by this API expire **15 minutes** after creation. The token lifetime cannot be extended. After expiration, a new token must be requested.

Parameters
None

Request Body:

```
{
  "kind": "request",
  "parameters": {
    "user": "xxxxxxxx",
    "password": "yyyyyyyy"
  }
}
```

© Copyright IBM Corp. 2025

Responses:

200: OK

```
{
  "kind": "response",
  "resource-name": "api-tokens",
  "resource-version": "1.0",
  "self": "/api/com.ibm.zaci.payload/api-tokens",
  "parameters": {
    "token": "sampletoken"
  }
}
```

Card management

GET /spyre-cards

This API returns a list of card IDs if `spyre_card_id` is not provided. If provided, returns detailed information for the specified card. It includes listing the available cards, getting a specific card details, activating a card and deactivating a card.

Parameters

API	Description
`spyre_card_id`	Provides the Spyre cards that are available on the system.
`spyre-cards?` `spyre_card_id=0000:00:00:1`	Provides the details of the specific Spyre card.
spyre-cards?spyre_card_id=all	Provides details of all the Spyre cards.

Responses:

200: Successful response

```
{
  "kind": "response",
  "parameters": [
    "string"
  ]
}
```

404: The provided URI does not map to an API operation.

500: An unexpected error has occurred during the API operation.

503: The API operation cannot be performed. Card management service is currently unavailable.

POST /spyre-cards/activate

This API activates a specific Spyre card by passing its ID in the query parameter or all associated Spyre cards by using `all` as the query parameter.

Parameters

spyre_card_id: ID of the Spyre card that is to be activated.

Responses:

202: Successful response

```
{
  "kind": "response",
  "parameters": {
    "self": "string"
  }
}
```

404: The provided URI does not map to an API operation.

500: An unexpected error has occurred during the API operation.

503: The API operation cannot be performed. Card management service is currently unavailable.

POST /spyre-cards/deactivate

This API is used to deactivate a single Spyre card. To deactivate a single Spyre card, pass the spyre_card_id as a query parameter. To deactivate all the Spyre cards pass "all" in query parameter.

Parameters

spyre_card_id: ID of the Spyre card that is to be deactivated.

Responses:

202: Successful response

```
{
  "kind": "response",
  "parameters": {
    "self": "string"
  }
}
```

404: The provided URI does not map to an API operation.

500: An unexpected error has occurred during the API operation.

503: The API operation cannot be performed. Card management service is unavailable.

Card configuration

DELETE /spyre-cards/senlib-images

This api removes a Senlib image on the SSA. The image is removed only if it is not in use by any Spyre Card on the Appliance. Admin needs to provide the image tag of Senlib/libVF to be deleted.

Parameters

tag: It is the ID of the container image that is to be deleted to the Spyre card.

Responses:

204: The API operation completed successfully, no response body is returned.

404: The provided URI does not map to an API operation.

409: The API operation cannot be performed, due to a temporary issue.

500: An unexpected error has occurred during the API operation.

503: The API operation cannot be performed. Card management service is currently unavailable.

GET /spyre-cards/senlib-images

This API is used to get available senlib images.

Parameters

None

Responses:

202: Successful response

```
{
  "kind": "response",
  "parameters": [
    "string"
  ]
}
```

404: The provided URI does not map to an API operation.

500: An unexpected error has occurred during the API operation.

503: The API operation cannot be performed. Card management service is currently unavailable.

PUT /spyre-cards/update-config

This API updates card configuration details such as pf, vf images and their log levels.

Parameters

spyre_card_id : ID of the spyre card that is to be updated.

Request body

```
{
  "kind": "request",
  "parameters": {
    "pftag": "string",
    "vftag": "string",
    "pfloglevel": "string",
    "vfloglevel": "string"
  }
}
```

Responses:

202: Successful response

```
{
  "kind": "response",
  "parameters": {
    "self": "string"
  }
}
```

404: The provided URI does not map to an API operation.

500: An unexpected error has occurred during the API operation.

503: The API operation cannot be performed. Card management service is unavailable.

Multi-Factor Authentication

These APIs manage and enforce multi-factor authentication (MFA) for user accounts. They allow the administrator to enable, disable, and perform first- and second-factor logins, as well as rotate or reset security secrets used in the authentication process.

This group of APIs supports:

- Enabling/disabling MFA: Toggle MFA for a given user based on credentials.
- First-factor login: Authenticate with username and password to receive a session for second-factor verification.
- Second-factor login: Authenticate using a second factor (such as TOTP) tied to the session from 1FA.
- Secret management: Rotate or reset server and reset secrets used to verify MFA tokens

Note:

1. Before enabling MFA, ensure that the system time is in sync with the UTC.
2. OTP is valid for one minute.

POST /auth/{username}/2fa

This endpoint is used to enable or disable Two-Factor Authentication (2FA) for a given user determined by the action query parameter, which must be either enable or disable.

Parameters

username: Enter the username of the admin

action: It can be enable or disable

Request body

```
{
  "kind": "request",
  "parameters": {
    "user": "username",
    "password": "password"
  }
}
```

Responses:

200: OK

```
{
  "kind": "request",
  "parameters": {
    "user": "username",
    "server_secret": "samplesecret",
    "reset_secret": "samplesecret"
  }
}
```

PUT /auth/{username}/2fa

This endpoint rotates or resets the secrets associated with 2FA either the server secret or the reset secret based on the action query parameter, which must be either rotate or reset.

Parameters

username: Enter the username

action: You can enter either rotate or reset.

Request body

```
{
  "kind": "request",
  "parameters": {
    "user": "username",
    "password": "password",
    "secret_type": "server",
    "totp": "sampleotp"
  }
}
```

Responses:

200: Rotate or reset the secrets (server secret or reset secret)

```
{
  "kind": "request",
  "parameters": {
    "user": "username",
    "server_secret": "samplesecret"
  }
}
```

GET /auth/{username}/2fa/status

This API retrieves the current two-factor authentication (2FA) status for the specified user. Use this endpoint to determine whether 2FA is currently enabled.

Parameters

username: Enter the username for which you want to retrieve the 2FA status.

Responses:

200: OK

```
{
  "kind": "request",
  "parameters": {
    "user": "username",
    "2fa_enabled": false
```

```
  }
}
```

POST /auth/{username}/login/1fa

This endpoint performs the first step of Two-Factor Authentication (1FA) by verifying the user's credentials

Parameters

username: Enter the username

Request body:

```
{
  "kind": "request",
  "parameters": {
    "user": "username",
    "password": "password"
  }
}
```

Responses:

200: Get session ID, required for the second-factor authentication.

```
{
  "kind": "request",
  "parameters": {
    "user": "username",
    "session_id": "samplesessionid"
  }
}
```

POST /auth/{username}/login/2fa

This endpoint performs the second step of Multi-Factor Authentication (2FA). It is used after successfully completing the first-factor login.

Parameters

username: Enter the username

Request body

```
{
  "kind": "request",
  "parameters": {
    "session_id": "samplesessionid",
    "totp": "sampleotp"
  }
}
```

Responses:

200: Get the session id, required for the second-factor login.

```
{
  "kind": "request",
  "parameters": {
    "user": "username",
    "token": "sampletoken"
  }
}
```

Certificate management

GET /certificates/v1

This API returns a collection of all certificates present in the system, including both issued and pending certificates.

Parameters

None

Responses:

200: A list of certificate instances


```
{
  "kind": "string",
  "self": "string",
  "resource-name": "string",
  "resource-version": "string",
  "instances": [
    {
      "serial": "string",
      "fingerprint": "string",
      "issued-to": "string",
      "issued-by": "string",
      "not-before": 0,
      "not-after": 0,
      "crt": "string",
      "csr": "string",
      "names": [
        "string"
      ],
      "state": "active",
      "hostname": "string",
      "ca": true,
      "id": "string"
    }
  ]
}
```


POST /certificates/v1

This API creates a new Certificate Signing Request (CSR) with the provided subject details and returns the certificate instance in CSR state.

Parameters

None

Request body


```
{
  "kind": "string",
  "resource-name": "string",
  "resource-version": "string",
  "parameters": {
    "ca": true,
    "c": "string",
    "o": "string",
    "ou": "string",
    "st": "string",
    "ip": "string"
  }
}
```


Responses:

201: CSR created


```
{
  "kind": "string",
  "self": "string",
  "resource-name": "string",
  "resource-version": "string",
  "properties": {
    "serial": "string",
    "fingerprint": "string",
    "issued-to": "string",
    "issued-by": "string",
    "not-before": 0,
    "not-after": 0,
    "crt": "string",
    "csr": "string",
    "names": [
      "string"
```

```
    ],
    "state": "active",
    "hostname": "string",
    "ca": true,
    "id": "string"
  }
}
```

DELETE /certificates/{certificateId}/v1

This API deletes the certificate identified by the certificate ID. Only certificates that are not active may be deleted.

Parameters

certificateId : ID of the certificate that is to be deleted.

Responses:

204: Deleted successfully

GET /certificates/{certificateId}/v1

This API retrieves the full details of a certificate using its unique certificate ID, including state, validity, subject, and issuer information.

Parameters

certificateId : ID of the certificate to retrieve detailed information.

Responses:

200: Certificate details

```
Certificate details

Media type

application/vnd.ibm.zaci.payload+json;version=1.0
Controls Accept header.
Example Value
Schema
{
  "kind": "string",
  "self": "string",
  "resource-name": "string",
  "resource-version": "string",
  "properties": {
    "serial": "string",
    "fingerprint": "string",
    "issued-to": "string",
    "issued-by": "string",
    "not-before": 0,
    "not-after": 0,
    "crt": "string",
    "csr": "string",
    "names": [
      "string"
    ],
    "state": "active",
    "hostname": "string",
    "ca": true,
    "id": "string"
  }
}
```

POST /certificates/{certificateId}/v1

This API performs a lifecycle action, such as self-sign, activate, deactivate on the specified certificate ID. It may optionally accept payload data.

Parameters

certificateId : ID of the certificate.

Responses:

200: Action succeeded, certificate instance returned.

```
{
  "kind": "string",
  "self": "string",
  "resource-name": "string",
  "resource-version": "string",
  "properties": {
    "serial": "string",
    "fingerprint": "string",
    "issued-to": "string",
    "issued-by": "string",
    "not-before": 0,
    "not-after": 0,
    "crt": "string",
    "csr": "string",
    "names": [
      "string"
    ],
    "state": "active",
    "hostname": "string",
    "ca": true,
    "id": "string"
  }
}
```

204: Action succeeded with no content.

PUT /certificates/{certificateId}/v1

This API is used to upload a certificate in plain text format to a specified certificate instance.

Parameters

certificateId : ID of the certificate.

Request body

```
-----BEGIN CERTIFICATE-----
...content of certificate...
-----END CERTIFICATE-----
```

Responses

200: Certificate uploaded successfully. The updated certificate instance is returned.

```
{
  "kind": "string",
  "self": "string",
  "resource-name": "string",
  "resource-version": "string",
  "properties": {
    "serial": "string",
    "fingerprint": "string",
    "issued-to": "string",
    "issued-by": "string",
    "not-before": 0,
    "not-after": 0,
    "crt": "string",
    "csr": "string",
    "names": [
      "string"
    ],
    "state": "active",
    "hostname": "string",
    "ca": true,
    "id": "string"
  }
}
```

404: The provided URI does not map to an API operation.

Diagnostic utilities

DELETE /spyre-cards/diag-utils

This API terminates any running diagnostic utility on a Spyre Card.

Parameters
spyre_card_id: It is the ID of the Spyre card

Responses:
204: The API operation is completed successfully, no response body is returned.

500: An unexpected error has occurred during the API operation.

503: The API operation cannot be performed. Card management service is currently unavailable.

GET /spyre-cards/diag-utils

This API gets the available diagnostic utilities on the Appliance.

Parameters
None

Responses:
200: Diagnostic utilities list

500: An unexpected error has occurred during the API operation.

503: The API operation cannot be performed. Card management service is currently unavailable.

POST /spyre-cards/diag-utils

This API runs diagnostics on the Spyre Card. The administrator can select a diagnostic utility from the set of available utilities provided within the Appliance.

Parameters
spyre_card_id: ID of the Spyre card on which the diagnostics should be run.

Request Body:
Content-Type: `application/json`

Responses:
202: Diagnostic run started

500: An unexpected error has occurred during the API operation.

503: The API operation cannot be performed. Card management service is unavailable.

GET /spyre-cards/diag-results

This API retrieves the results of a previously run diagnostic utility. The logs are compressed into a `.zip` file and returned in the response.

Parameters
spyre_card_id: ID of the Spyre card for which the details are to be retrieved.

Responses:
200: Successfully collected diagnostics results

500: An unexpected error has occurred during the API operation.

503: The API operation cannot be performed. Card management service is unavailable.

Logs and Telemetry

GET /spyre-cards/logs

This API fetches the Spyre card logs. This includes physical function and virtual function logs of a specific card

Parameters

spyre_card_id: ID of the spyre card to retrieve logs.

Responses:

200: Logs archive

500: An unexpected error has occurred during the API operation.

GET /spyre-cards/logs/card_mgmt

This API fetches the card management logs.

Parameters

spyre_card_id: ID of the spyre card to retrieve logs.

Responses:

200: Successfully collected card management logs.

500: An unexpected error has occurred during the API operation.

GET /spyre-cards/logs/all

This API fetches all the logs related to card management and senlib/libVF containers. These logs will be compressed into a tar file and returned.

Parameters

None

Responses:

200: Successfully gathered the card logs.

500: An unexpected error has occurred during the API operation.

GET /spyre-cards/logs/flight_logs

This API fetches the flight logs of the card. These logs will be compressed into a tar file and returned.

Parameters

None

Responses:

200: Successfully gathered the card logs.

500: An unexpected error has occurred during the API operation.

503: The API operation cannot be performed. Card management service is currently unavailable.

POST /spyre-cards/logs/flight_logs

This API collects flight logs on the Spyre Card.

Parameters

spyre_card_id: ID of the spyre card for which flight logs are to be collected.

Responses:

200: Flight logs collection started.

500: An unexpected error has occurred during the API operation.

503: The API operation cannot be performed. Card management service is currently unavailable.

GET /spyre-cards/telemetry

This API gathers the available telemetry data. Current metrics data is for temperature and power. This file will be compressed and returned as a tar file.

Parameters

None

Responses:

200: Collected telemetry data.

500: An unexpected error has occurred during the API operation.

503: The API operation cannot be performed. Card management service is currently unavailable.

RAS Events

This API describes the Spyre card resource and its operations for querying RAS events related to Spyre cards.

GET /spyre-cards/ras

This API retrieves Reliability, Availability, Serviceability (RAS) events based on various filters such as card ID, category, action, and time range.

Parameters

Parameter	Description
spyre_card_id	ID of the Spyre card.
from	Start time (format: `YYYY-MM-DDTHH:MM:SS`)
to	End time (format: `YYYY-MM-DDTHH:MM:SS`)
category	`hardware`, `software`, `linuxconfig`, `configuration`, or `all`
action	`reset`, `information`, `deconfigure`, `configure`, `recoverable_event`, `telemetry`, `No action`
code	This is a hexadecimal value which is unique for a particular RAS event. Ex: `0x386b`

Note: SSA supports fetching RAS events for a maximum period of one week. If no query parameters are provided in the API request, it returns all RAS events that occurred within the last seven days.

Request body

```
curl -k -X GET https://localhost/api/com.ibm.zaci.system/spyre-cards/ras \
  -H "Authorization: Bearer $token" \
  -H "zACI-API:com.ibm.zaci.system/1.0" \
  -H "Accept: application/vnd.ibm.zaci.payload+json;version=1.0" \
  -H "Content-Type: application/vnd.ibm.zaci.payload+json;version=1.0" \
  -o ras.tar
```

Responses:

200: Successful response. It returns a `.tar` file containing RAS events in JSON format

Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprise is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation, in the United States and/or other countries. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on http://www.ibm.com/trademark.

© Copyright IBM Corp. 2025

Class A Notices

The following Class A statements apply to this IBM product. The statement for other IBM products intended for use with this product will appear in their accompanying manuals.

Federal Communications Commission (FCC) Statement

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

Properly shielded and grounded cables and connectors must be used in order to meet FCC emission limits. IBM is not responsible for any radio or television interference caused by using other than recommended cables and connectors or by unauthorized changes or modifications to this equipment. Unauthorized changes or modifications could void the user's authority to operate the equipment.

This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Industry Canada Compliance Statement

This Class A digital apparatus complies with Canadian ICES-003.

United Kingdom Notice

This product may cause interference if used in residential areas. Such use must be avoided unless the user takes special measures to reduce electromagnetic emissions to prevent interference to the reception of radio and television broadcasts.

Avis de conformité à la réglementation d'Industrie Canada

Cet appareil numérique de la classe A est conforme à la norme NMB-003 du Canada.

European Community Compliance Statement

This product is in conformity with the protection requirements of EU Council Directive 2014/30/EU on the approximation of the laws of the Member States relating to electromagnetic compatibility. IBM cannot accept responsibility for any failure to satisfy the protection requirements resulting from a non-recommended modification of the product, including the fitting of non-IBM option cards.

This product has been tested and found to comply with the limits for Class A Information Technology Equipment according to European Standard EN 55032. The limits for Class A equipment were derived for commercial and industrial environments to provide reasonable protection against interference with licensed communication equipment.

European Community contact:
IBM Deutschland GmbH
Technical Regulations, Department M372
IBM-Allee 1, 71139 Ehningen, Germany
Tele: +49 (0) 800 225 5423 or +49 (0) 180 331 3233
email: halloibm@de.ibm.com

Warning: This is a Class A product. In a domestic environment, this product may cause radio interference, in which case the user may be required to take adequate measures.

Japan Voluntary Control Council for Interference (VCCI) Notice

この装置は、クラス A 機器です。 この装置を住宅環境で使用すると電波妨害を引き起こすことがあります。 この場合には使用者が適切な対策を講ずるよう要求される ことがあります。 ＶＣＣＩ－Ａ

The following is a summary of the Japanese VCCI statement above:

This is a Class A product based on the standard of the VCCI Council. If this equipment is used in a domestic environment, radio interference may occur, in which case the user may be required to take corrective actions.

Japan Electronics and Information Technology Industries Association (JEITA) Notice

（一社）電子情報技術産業協会　高調波電流抑制対策実施
要領に基づく定格入力電力値：IBM Documentationの各製品の仕様ページ参照

This statement applies to products less than or equal to 20 A per phase.

高調波電流規格　JIS C 61000-3-2 適合品

These statements apply to products greater than 20 A, single-phase.

高調波電流規格　JIS C 61000-3-2 準用品

本装置は、「高圧又は特別高圧で受電する需要家の高調波抑制対策ガイドライン」対象機器（高調波発生機器）です。
回路分類：6（単相、ＰＦＣ回路付）
換算係数：0

These statements apply to products greater than 20 A per phase, three-phase.

高調波電流規格　JIS C 61000-3-2 準用品

本装置は、「高圧又は特別高圧で受電する需要家の高調波抑制対策ガイドライン」対象機器（高調波発生機器）です。
回路分類　：5（3相、ＰＦＣ回路付）
換算係数　：0

People's Republic of China Notice

警告:在居住环境中,运行此设备可能会造成无线电干扰。

Declaration: This is a Class A product. In a domestic environment, this product may cause radio interference, in which case the user may need to perform practical action.

Taiwan Notice

CNS 13438:

警告使用者：
此為甲類資訊技術設備，
於居住環境中使用時，
可能會造成射頻擾動，在此種情況下，
使用者會被要求採取某些適當的對策。

CNS 15936:

警告：為避免電磁干擾，本產品不應安裝或使用於住宅環境。

IBM Taiwan Contact Information:

台灣IBM 產品服務聯絡方式：
台灣國際商業機器股份有限公司
台北市松仁路7號3樓
電話：0800-016-888

Electromagnetic Interference (EMI) Statement - Korea

이 기기는 업무용(A급)으로 전자파적합등록을 한 기기이오니
판매자 또는 사용자는 이 점을 주의하시기 바라며, 가정외의
지역에서 사용하는 것을 목적으로 합니다.

Germany Compliance Statement

Deutschsprachiger EU Hinweis: Hinweis für Geräte der Klasse A EU-Richtlinie zur Elektromagnetischen Verträglichkeit

Dieses Produkt entspricht den Schutzanforderungen der EU-Richtlinie 2014/30/EU zur Angleichung der Rechtsvorschriften über die elektromagnetische Verträglichkeit in den EU-Mitgliedsstaaten und hält die Grenzwerte der EN 55032 Klasse A ein.

Um dieses sicherzustellen, sind die Geräte wie in den Handbüchern beschrieben zu installieren und zu betreiben. Des Weiteren dürfen auch nur von der IBM empfohlene Kabel angeschlossen werden. IBM übernimmt keine Verantwortung für die Einhaltung der Schutzanforderungen, wenn das Produkt ohne Zustimmung von IBM verändert bzw. wenn Erweiterungskomponenten von Fremdherstellern ohne Empfehlung von IBM gesteckt/eingebaut werden.

EN 55032 Klasse A Geräte müssen mit folgendem Warnhinweis versehen werden:

"Warnung: Dieses ist eine Einrichtung der Klasse A. Diese Einrichtung kann im Wohnbereich Funk-Störungen verursachen; in diesem Fall kann vom Betreiber verlangt werden, angemessene Maßnahmen zu ergreifen und dafür aufzukommen."

Deutschland: Einhaltung des Gesetzes über die elektromagnetische Verträglichkeit von Geräten

Dieses Produkt entspricht dem "Gesetz über die elektromagnetische Verträglichkeit von Geräten (EMVG)". Dies ist die Umsetzung der EU-Richtlinie 2014/30/EU in der Bundesrepublik Deutschland.

Zulassungsbescheinigung laut dem Deutschen Gesetz über die elektromagnetische Verträglichkeit von Geräten (EMVG) (bzw. der EMC EG Richtlinie 2014/30/EU) für Geräte der Klasse A

Dieses Gerät ist berechtigt, in Übereinstimmung mit dem Deutschen EMVG das EG-Konformitätszeichen - CE - zu führen.

Verantwortlich für die Einhaltung der EMV Vorschriften ist der Hersteller:
International Business Machines Corp.
New Orchard Road
Armonk, New York 10504
Tel: 914-499-1900

Der verantwortliche Ansprechpartner des Herstellers in der EU ist:
IBM Deutschland GmbH
Technical Regulations, Abteilung M372
IBM-Allee 1, 71139 Ehningen, Germany
Tel: +49 (0) 800 225 5423 or +49 (0) 180 331 3233
email: halloibm@de.ibm.com

Generelle Informationen:

Das Gerät erfüllt die Schutzanforderungen nach EN 55024 und EN 55032 Klasse A.

Electromagnetic Interference (EMI) Statement - Russia

ВНИМАНИЕ! Настоящее изделие относится к классу А.
В жилых помещениях оно может создавать радиопомехи, для снижения которых необходимы дополнительные меры

Electromagnetic Interference (EMI) Statement - Kingdom of Saudi Arabia Notice

قد يتسبب هذا المنتج في حدوث تداخل إذا تم استخدامه في المناطق السكنية.

ويجب تجنب هذا الاستخدام ما لم يتخذ المستخدم تدابير خاصة لتقليل الانبعاثات الكهرومغناطيسية لمنع التداخل مع استقبال البث الإذاعي والتلفزيوني.

تحذير: هذا الجهاز متوافق مع الفئة أ من SASO CISPR 32

في البيئة السكنية، قد يتسبب هذا الجهاز في حدوث تداخل لاسلكي.

GC28-7072-00