# About This Document

## Purpose

Implement a REST based command line utility for IBM Guardium Key Lifecycle Manager

## Scope

The scope of this project will be to create a command line interface that gives users the ability to run all the commands available in the current version of GKLM through Swagger.

The utility will require the cert from Websphere to work. The cert should be put in the directory where the utility is run from.

The utility is being tested on Windows and Linux platforms, but should run anywhere Java can be run.

# Overview

## Personas

SKLM users and administrators will be the users of this utility.

## Use Case Scenarios

There are many use cases for this utility. At a high level, the use cases are to be able to run each REST command that is available through the Swagger interface in the current version of GKLM.

Examples:

* User needs to create a new certificate in GKLM (/SKLM/rest/v1/certificates)
* User want to list keys in GKLM (/SKLM/rest/v1/keys)
* User wants to run a GKLM backup (/SKLM/rest/v1/ckms/backups)

## Design Goals and Constraints

The main design goal is to create a utility that is portable across Windows and Linux that can be easily maintained and expanded when new functionality is available. We are also hoping to make this code available via the web for other users to contribute their own scripts that utilize our code.

## Design Requirements

* Provide a mapping between a Requirement from the Requirements document and how that requirement is covered by this Design.  A table can be used to depict this mapping e.g.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Not Applicable |  |  |

## Assumptions and Dependencies

There is an assumption that the error messages returned from the SKLM server when interacting with REST are sufficient for our utility to use.

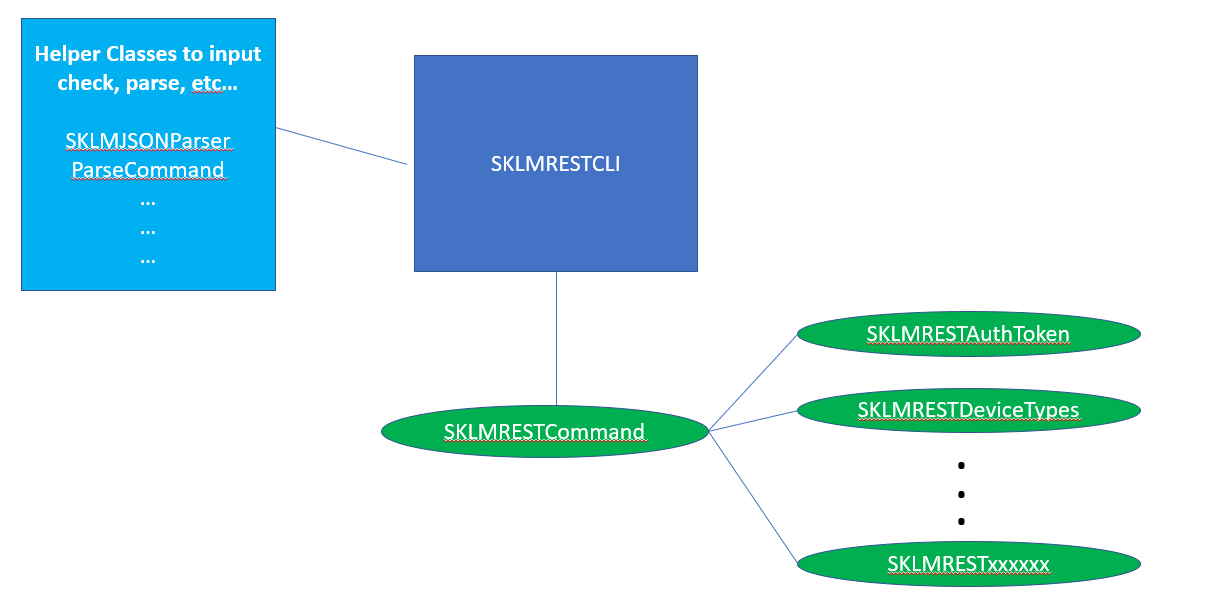
There may be some dependencies on Java packages that are readily available. The utility will be kept as ‘Java native’ as possible.

## Delivery Plan

There are no current discussions on breaking this into phases, although a subset of the total number of commands could be delivered in phases if needed.

# High Level Design

## Architecture



Along with the helper files that do things like read in data, etc, the main class that takes input will be SKLMRESTCLI. When running a command, each of the command will have their own class, but they will be extended from the SKLMRESTCommand class which will have common functionality for all commands contained within.

A python script called **swagger.py** can be run with the swagger\_en.yaml file in the same directory, and it will produce a file called ParsedSKLMYaml.txt that is required/consumed by the utility. It is only necessary to run this if additional commands are added to the swagger file.

The format of ParsedSKLMYaml.txt is:

Path:

Type: {post,get,delete,put}

Summary:

Attributes:

RequiredAttribute:

AttributeLocs:

.

.

.

--END--

There is a single Path that represents the command, and then sections that consist of Type, Summary, Attributes, Required Attributes, and Attribute Locations for each 'type' the command can use. The end of a command is marked by the string --END--. Summary is a text string summary of the command and the other 3 sections are comma separated lists that represent the respective label. As an example:

Path:/SKLM/rest/v1/ckms/deviceGroupsExport

Type:post

Summary:To create an encrypted archive that contains device group data for exporting to another instance of IBM Security Guardium Key Lifecycle Manager.

Attributes:Accept-Language,exportDirectory,password,name,description

RequiredAttributes:name,password

AttributeLocs:header,body,body,body,body

Type:get

Summary:To list exported files in a directory.

Attributes:Accept-Language,exportDirectory

RequiredAttributes:

AttributeLocs:header,query

--END--

The above example command can be used via the post or get methods. In the get version, there are no required attributes, but there are 2 optional attributes. If specified, 'Accept-Language' will go in the header of the request and 'exportDirectory' will be put on the query line.

The main interface will have helper classes that do things like parse commands, messages, etc. It will take the command and pass the necessary data to other classes that implement each REST command.

The expected flow will be to run the main class with:

* The REST URL - optional
* Username (SKLMAdmin)
* Password
* Certificate file
* Command to be run in quotes – optional
* Path to a file with commands to be run in order on separate lines - optional

The first thing the program will do is to get an authentication token and then present a prompt where the user can enter commands of the form:

command(arg1=val1,arg2=val2….)

for example:

sklmAuthToken(userid=SKLMAdmin,password=Passw0rd\_1)

The values can optionally be put in quotes. If a value has a comma in it, you must put it in quotes.

The actual commands will be documented as we go since we have to allow for some commands to be run as post, put, etc.

The user can also enter ‘help’ or ‘?’ on the command line to get a list of supported commands, and for each command they can use an entry of the form command(help) to get more information on the command.

The following functionality will be implemented (to be verified against what is available in 4.1 currently):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Login** |  | | POST | ​/SKLM​/rest​/v1​/ckms​/login | | DELETE | ​/SKLM​/rest​/v1​/ckms​/logout | |
| |  |  | | --- | --- | | **Conflict resolution** | | | POST | ​/SKLM​/rest​/v1​/ckms​/conflictResolution​/changeName | | POST | ​/SKLM​/rest​/v1​/ckms​/conflictResolution​/changeCertificateAlias | | POST | ​/SKLM​/rest​/v1​/ckms​/conflictResolution​/renewKeyAlias | | POST | ​/SKLM​/rest​/v1​/ckms​/conflictResolution​/renewUUID | | GET | ​/SKLM​/rest​/v1​/ckms​/conflictResolution​/getChangeHistory | |
| |  |  | | --- | --- | | **Key group management** | | | POST | ​/SKLM​/rest​/v1​/keygroups​/{group} | | DELETE | ​/SKLM​/rest​/v1​/keygroups​/{group} | | POST | ​/SKLM​/rest​/v1​/keygroupentry | | DELETE | ​/SKLM​/rest​/v1​/keygroupentry | | POST | ​/SKLM​/rest​/v1​/keygroups​/rollover | | GET | ​/SKLM​/rest​/v1​/keygroups​/rollover | | GET | ​/SKLM​/rest​/v1​/keygroups | | PUT | ​/SKLM​/rest​/v1​/keygroups | |
| |  |  | | --- | --- | | **Device management** | | | POST | ​/SKLM​/rest​/v1​/machines​/device | | DELETE | ​/SKLM​/rest​/v1​/machines​/device | | GET | ​/SKLM​/rest​/v1​/machines​/device | | DELETE | ​/SKLM​/rest​/v1​/devices​/{uuid} | | POST | ​/SKLM​/rest​/v1​/machines | | DELETE | ​/SKLM​/rest​/v1​/machines | | GET | ​/SKLM​/rest​/v1​/machines | | PUT | ​/SKLM​/rest​/v1​/machines | | POST | ​/SKLM​/rest​/v1​/devices | | GET | ​/SKLM​/rest​/v1​/devices | | PUT | ​/SKLM​/rest​/v1​/devices | |
| |  |  | | --- | --- | | **Pending object management** | | | GET | ​/SKLM​/rest​/v1​/pendingMachineDevices | | POST | ​/SKLM​/rest​/v1​/pendingDevices | | GET | ​/SKLM​/rest​/v1​/pendingDevices | | DELETE | ​/SKLM​/rest​/v1​/pendingDevices​/{uuid} | | DELETE | ​/SKLM​/rest​/v1​/pendingClientCertificates​/{uuid} | | POST | ​/SKLM​/rest​/v1​/pendingMachineDevices​/accept | | GET | ​/SKLM​/rest​/v1​/pendingClientCertificates | | POST | ​/SKLM​/rest​/v1​/pendingClientCertificates | | POST | ​/SKLM​/rest​/v1​/pendingMachineDevices​/reject | |
| |  |  | | --- | --- | | **Device group management** | | | GET | ​/SKLM​/rest​/v1​/deviceTypes | | POST | ​/SKLM​/rest​/v1​/ckms​/deviceGroupsExport​/viewSummaryOfExportFileInADirectory | | POST | ​/SKLM​/rest​/v1​/ckms​/deviceGroupsExport | | GET | ​/SKLM​/rest​/v1​/ckms​/deviceGroupsExport | | GET | ​/SKLM​/rest​/v1​/deviceGroups​/base | | POST | ​/SKLM​/rest​/v1​/ckms​/deviceGroupsImport​/importConflicts | | DELETE | ​/SKLM​/rest​/v1​/ckms​/deviceGroupsExport​/deleteExportFile | | POST | ​/SKLM​/rest​/v1​/ckms​/deviceGroupsImport | | GET | ​/SKLM​/rest​/v1​/deviceGroups | | GET | ​/SKLM​/rest​/v1​/deviceTypes​/{deviceListType} | | POST | ​/SKLM​/rest​/v1​/deviceGroups​/{name} | | DELETE | ​/SKLM​/rest​/v1​/deviceGroups​/{name} | | GET | ​/SKLM​/rest​/v1​/deviceGroupAttributes | | PUT | ​/SKLM​/rest​/v1​/deviceGroupAttributes | | DELETE | ​/SKLM​/rest​/v1​/deviceGroupAttributes | |
| |  |  | | --- | --- | | **Multi-Master cluster management** | | | POST | ​/SKLM​/rest​/v1​/ckms​/multimaster​/stopagent | | GET | ​/SKLM​/rest​/v1​/ckms​/config​/nodes​/joinBackTheCluster | | POST | ​/SKLM​/rest​/v1​/ckms​/config​/nodes​/removeNode | | GET | ​/SKLM​/rest​/v1​/ckms​/config​/nodes​/isNodeIsolatedFromCluster | | POST | ​/SKLM​/rest​/v1​/ckms​/config​/nodes​/addNodes | | GET | ​/SKLM​/rest​/v1​/ckms​/nodes​/getThisNodeDetails | | POST | ​/SKLM​/rest​/v1​/ckms​/config​/nodes​/restartCluster | | POST | ​/SKLM​/rest​/v1​/ckms​/nodes​/checkPreRequisite | | GET | ​/SKLM​/rest​/v1​/ckms​/nodes | | GET | ​/SKLM​/rest​/v1​/ckms​/config​/nodes​/setupAsReadWriteMaster | | POST | ​/SKLM​/rest​/v1​/ckms​/config​/nodes​/takeoverAsPrimary | | GET | ​/SKLM​/rest​/v1​/ckms​/nodes​/getClusterHADRStatus | | GET | ​/SKLM​/rest​/v1​/ckms​/nodes​/allNodeStatus | | POST | ​/SKLM​/rest​/v1​/ckms​/multimaster​/startagent | | POST | ​/SKLM​/rest​/v1​/ckms​/config​/nodes​/stopCluster | | GET | ​/SKLM​/rest​/v1​/ckms​/multimaster​/agentstatus | |
| |  |  | | --- | --- | | **Server management** | | | GET | ​/SKLM​/rest​/v1​/keyServerStatus | | GET | ​/SKLM​/rest​/v1​/ldap | | POST | ​/SKLM​/rest​/v1​/ldap | | POST | ​/SKLM​/rest​/v1​/ldap​/addLDAPUserToGroup | | GET | ​/SKLM​/rest​/v1​/ckms​/versionInfo | | GET | ​/SKLM​/rest​/v1​/license | | GET | ​/SKLM​/rest​/v1​/ckms​/counts | | GET | ​/SKLM​/rest​/v1​/ckms​/counts​/{countType} | | POST | ​/SKLM​/rest​/v1​/ckms​/servermanagement​/restartServer | | GET | ​/SKLM​/rest​/v1​/health | | PUT | ​/SKLM​/rest​/v1​/ckms​/changePassword​/db2​/multimaster | | GET | ​/SKLM​/rest​/v1​/systemDetails | | PUT | ​/SKLM​/rest​/v1​/ckms​/changePassword​/db2​/standalone | | GET | ​/SKLM​/rest​/v1​/systemConfigProperties | | POST | ​/SKLM​/rest​/v1​/systemConfigProperties | |
| |  |  | | --- | --- | | **Served data archival** | | | POST | ​/SKLM​/rest​/v1​/archive​/servedData | | GET | ​/SKLM​/rest​/v1​/servedData | |
| |  |  | | --- | --- | | **KMIP secret data management** | | | GET | ​/SKLM​/rest​/v1​/secretData | | DELETE | ​/SKLM​/rest​/v1​/secretData​/{uuid} | |
| |  |  | | --- | --- | | **Certificate management** | | | POST | ​/SKLM​/rest​/v1​/certificates​/rollover | | GET | ​/SKLM​/rest​/v1​/certificates​/rollover | | DELETE | ​/SKLM​/rest​/v1​/certificates​/rollover​/{uuid} | | PUT | ​/SKLM​/rest​/v1​/certificates​/bulkCertUpdate | | GET | ​/SKLM​/rest​/v1​/certificates | | POST | ​/SKLM​/rest​/v1​/certificates | | PUT | ​/SKLM​/rest​/v1​/certificates | | PUT | ​/SKLM​/rest​/v1​/certificateAttributes | | POST | ​/SKLM​/rest​/v1​/certificates​/import | | DELETE | ​/SKLM​/rest​/v1​/certificates​/{alias} | | PUT | ​/SKLM​/rest​/v1​/certificates​/client | | PUT | ​/SKLM​/rest​/v1​/certificates​/export | |
| |  |  | | --- | --- | | **Key management** | | | GET | ​/SKLM​/rest​/v1​/keys​/integrityCheck | | POST | ​/SKLM​/rest​/v1​/keys​/import | | POST | ​/SKLM​/rest​/v1​/keys | | GET | ​/SKLM​/rest​/v1​/keys | | PUT | ​/SKLM​/rest​/v1​/keys | | DELETE | ​/SKLM​/rest​/v1​/keys​/{alias} | | PUT | ​/SKLM​/rest​/v1​/keys​/export | |
| |  |  | | --- | --- | | **Backup and restore** | | | GET | ​/SKLM​/rest​/v1​/ckms​/backups​/result | | GET | ​/SKLM​/rest​/v1​/ckms​/restore​/progress | | GET | ​/SKLM​/rest​/v1​/ckms​/restore​/result | | GET | ​/SKLM​/rest​/v1​/ckms​/backups​/isrunning | | POST | ​/SKLM​/rest​/v1​/ckms​/backups | | GET | ​/SKLM​/rest​/v1​/ckms​/backups | | GET | ​/SKLM​/rest​/v1​/ckms​/restore​/isrunning | | GET | ​/SKLM​/rest​/v1​/backupInfo | | POST | ​/SKLM​/rest​/v1​/ckms​/restore | | GET | ​/SKLM​/rest​/v1​/ckms​/backups​/progress | | GET | ​/SKLM​/rest​/v1​/ckms​/backups​/need | |
| |  |  | | --- | --- | | **Master key management** | | | POST | ​/SKLM​/rest​/v1​/ckms​/deviceGroupMasterKey​/{deviceGroupName} | | GET | ​/SKLM​/rest​/v1​/ckms​/deviceGroupMasterKey​/{devicegroupName} | | POST | ​/SKLM​/rest​/v1​/ckms​/deviceGroupMasterKey​/{devicegroupName}​/rotate | | POST | ​/SKLM​/rest​/v1​/ckms​/deviceGroupMasterKey​/rotate | | POST | ​/SKLM​/rest​/v1​/ckms​/masterKey | | POST | ​/SKLM​/rest​/v1​/ckms​/masterKey​/transmitter | | GET | ​/SKLM​/rest​/v1​/ckms​/deviceGroupMasterKey | |
| |  |  | | --- | --- | | **Replication** |  | | GET | ​/SKLM​/rest​/v1​/replicationConfigProperties | | PUT | ​/SKLM​/rest​/v1​/replicationConfigProperties | | DELETE | ​/SKLM​/rest​/v1​/replicationConfigProperties | | POST | ​/SKLM​/rest​/v1​/replicate​/now | | POST | ​/SKLM​/rest​/v1​/replicate​/stop | | POST | ​/SKLM​/rest​/v1​/replicate​/start | | GET | ​/SKLM​/rest​/v1​/replicate​/status | | GET | ​/SKLM​/rest​/v1​/replicationConfigProperties​/{propertyName} | |
| |  |  | | --- | --- | | **Truststore management** | | | GET | ​/SKLM​/rest​/v1​/trustStoreCertificates | | PUT | ​/SKLM​/rest​/v1​/trustStoreCertificates​/addCertToTrustStore | | PUT | ​/SKLM​/rest​/v1​/trustStoreCertificates​/deleteCertFromTrustStore | |
| |  |  | | --- | --- | | **Object management** | | | POST | ​/SKLM​/rest​/v1​/objects​/secret | | POST | ​/SKLM​/rest​/v1​/objects​/keypair | | POST | ​/SKLM​/rest​/v1​/objects​/opaque | | GET | ​/SKLM​/rest​/v1​/objects | | GET | ​/SKLM​/rest​/v1​/objects​/{objectId} | | DELETE | ​/SKLM​/rest​/v1​/objects​/{objectId} | | POST | ​/SKLM​/rest​/v1​/objects​/symmetrickey | | POST | ​/SKLM​/rest​/v1​/objects​/certificate | |
| |  |  | | --- | --- | | **Client management** | | | PUT | ​/SKLM​/rest​/v1​/clients​/{clientName}​/assignUsers | | PUT | ​/SKLM​/rest​/v1​/clients​/updateClientName | | PUT | ​/SKLM​/rest​/v1​/clients​/{clientName}​/assignCertificate | | DELETE | ​/SKLM​/rest​/v1​/clients​/{clientName} | | GET | ​/SKLM​/rest​/v1​/clients​/{clientName} | | POST | ​/SKLM​/rest​/v1​/clients | | PUT | ​/SKLM​/rest​/v1​/clients​/{clientName}​/removeUsers | | GET | ​/SKLM​/rest​/v1​/clients​/{clientName}​/listObjectsToClient | |
| |  |  |  | | --- | --- | --- | | **File transfer** | |  | | POST | ​/SKLM​/rest​/v1​/filetransfer​/upload​/license | | | GET | ​/SKLM​/rest​/v1​/filetransfer​/download​/objectfiles | | | POST | ​/SKLM​/rest​/v1​/filetransfer​/upload​/objectfiles | | | GET | ​/SKLM​/rest​/v1​/filetransfer​/download​/logs | | |
| |  |  | | --- | --- | | **KMIP template management** | | | DELETE | ​/SKLM​/rest​/v1​/kmipTemplate​/{uuid} | | GET | ​/SKLM​/rest​/v1​/kmipTemplate | |
| |  |  | | --- | --- | | **Server configuration** | | | GET | ​/SKLM​/rest​/v1​/configProperties | | PUT | ​/SKLM​/rest​/v1​/configProperties | | DELETE | ​/SKLM​/rest​/v1​/configProperties | | GET | ​/SKLM​/rest​/v1​/configProperties​/{propertyName} | |

## Management / Admin Interfaces

SKLMRESTCLI is the admin interface. When run with the correct parameters, it will give an output similar to what wsadmin does today:

sklmRestCLI>

where commands can be input or ‘quit’ can be entered to quit out of the program. There will be an option to add a command to be run, so no user interaction is required if the command is input correctly. The user can also enter ‘help’ or ‘?’ to get a list of available commands and something of the form *command*(help) to get info on a command.

## End User Interfaces

The end user interface is the same as above.

## Logs and Audit

### Audit

Not applicable

### Logging and Tracing

The command window and Log4j will be used for logging. Any output that would fit in a log file better than the window such as debug message will be logged to a file according to the following log4j configuration:

<?xml version="1.0" encoding="UTF-8"?>

<Configuration status="ERROR">

<Appenders>

<File name="file" fileName="/tmp/sklmrestcli.out">

<PatternLayout>

<pattern>%d{dd MMM yyyy HH:mm:ss,SSS} [%c{1}] %p - %m%n</pattern>

</PatternLayout>

</File>

</Appenders>

<Loggers>

<Root level="DEBUG" additivity="false">

<AppenderRef ref="file" level="DEBUG"/>

</Root>

</Loggers>

</Configuration>

## Reporting and Notifications

Not applicable

## Security Considerations

Initially, it was thought the program would utilize an ‘empty truststore’ in java, meaning that it would just take whatever certificate the server sends and treat it as trusted.

We have since added the ability for a certificate file to be used to verify the server cert.

## Regulatory Compliance Considerations

Not applicable

## Integration

Not applicable

## Deprecation and Migration

Although not a requirement, this interface could/should deprecate the wsadmin interface. The only requirement is that users familiarize themselves with the format of the new command interface.

## Future Considerations

Initial version

## Discarded Technical Options

None

## Performance

It is not expected that this utility will add any significant overhead to the system.

## Accessibility

I wasn’t able to find much of anything about accessibility for a command line tool in the guidelines.

## Globalization

For the first version, I was making an assumption our output would be English only. I am unsure on whether or not the current Swagger interface will return things in other languages?

## References

REST APIs - <https://www.ibm.com/docs/en/sgklm/4.1?topic=reference-rest-apis>

wsadmin interface - <https://www.ibm.com/docs/en/sgklm/4.1?topic=reference-command-line-interface>

# Operations Consideration (Non Functional Requirements)

## Operational Architecture

Not applicable

## Deployment (Installation and Configuration)

Utility may be installed with the next release but will also be available for download. The ‘install’ should just be extracting a zip file to a new directory or over and existing one.

## Automation

Any type of deployment, upgrade, or rollback should be able to be accomplished with an extraction of the utility over the current installation.

## Security

Any security patches should be able to be accomplished with an extraction of the utility over the current installation.

## Health check Monitoring

There is no health checking slated to be built into this utility.

## Multi-Tenancy

This utility will be inherently multi-tenant. Multiple users could run multiple instances of the utility.

## High Availability

High Availability is not an aspect that will be incorporated into this utility.

## Disaster recovery

Not applicable

## Performance and Scalability

Performance if the utility is largely dependent on the network and the internals of processing existing commands. And performance/scaling concerns are negligible.

## Capacity Planning

The memory and storage consumed by this utility should be negligible.

# Open Source or Third Party Software

* javax.json-1.0.2.jar->Upgraded to javax.json-1.1.4.jar
* disruptor-3.3.4.jar 🡨 I am not sure we need this one
* log4j-api-2.13.3.jar
* log4j-core-2.13.3.jar

# Test Considerations

All the commands should be verified in both success and failure scenarios.

As many permutations as possible for options on each command should be tested.

The interface and its options should be tested for all valid types of input as well as several incorrect command scenarios.

# Documentation

Documentation would be written for the end user on how to get into the interface and different scenarios of running commands.

Examples of a few commands will be given while also pointing them to the existing documentation for existing REST commands. This should eliminate the need to update documentation for the utility when new function is added.

Videos of the examples being run may also be provided.