Feature Specification Template

	*Project Name: **GOLang library for HPE iLO Restful Operations.***
	*Author Names: **Aman Narayan Singh, Swathi BJ, Aditya Mishra, Shreesh Kulkarni*** *Feature Specification Document* *Date: 19/06/2022*
	<Version 1.1>

Revision History

Version No.	***Version Date***	***Revised By***	***Reviewed By***	***Approved By***	***Affected section and description of change***
1.0	*20/06/2022*	*Aman Narayan Singh*	*Kallur Rajeevalochana.*		*The first version contained a design about server side which has to be updated to client side in the next version.*

1.1	*26/06/2022*	*Aman Narayan Singh ,* *Swathi BJ*	*Kallur Rajeevalochana*		*To be reviewed on 29/07/2022* *Updated to client side program which can carry out the GET operation with iLO restful operations.*

Contents

Background

Purpose

HPE is a leader in server software which comprises Integrated Lights out(iLO). To manage iLO using Restful Redfish APIs, the Golang library is the first step. Library should be independently used to develop new Golang tools to manage HPE servers.

Scope

<Systems, Users, Models that is in the scope of the feature or change request. These need to be outlined here.>

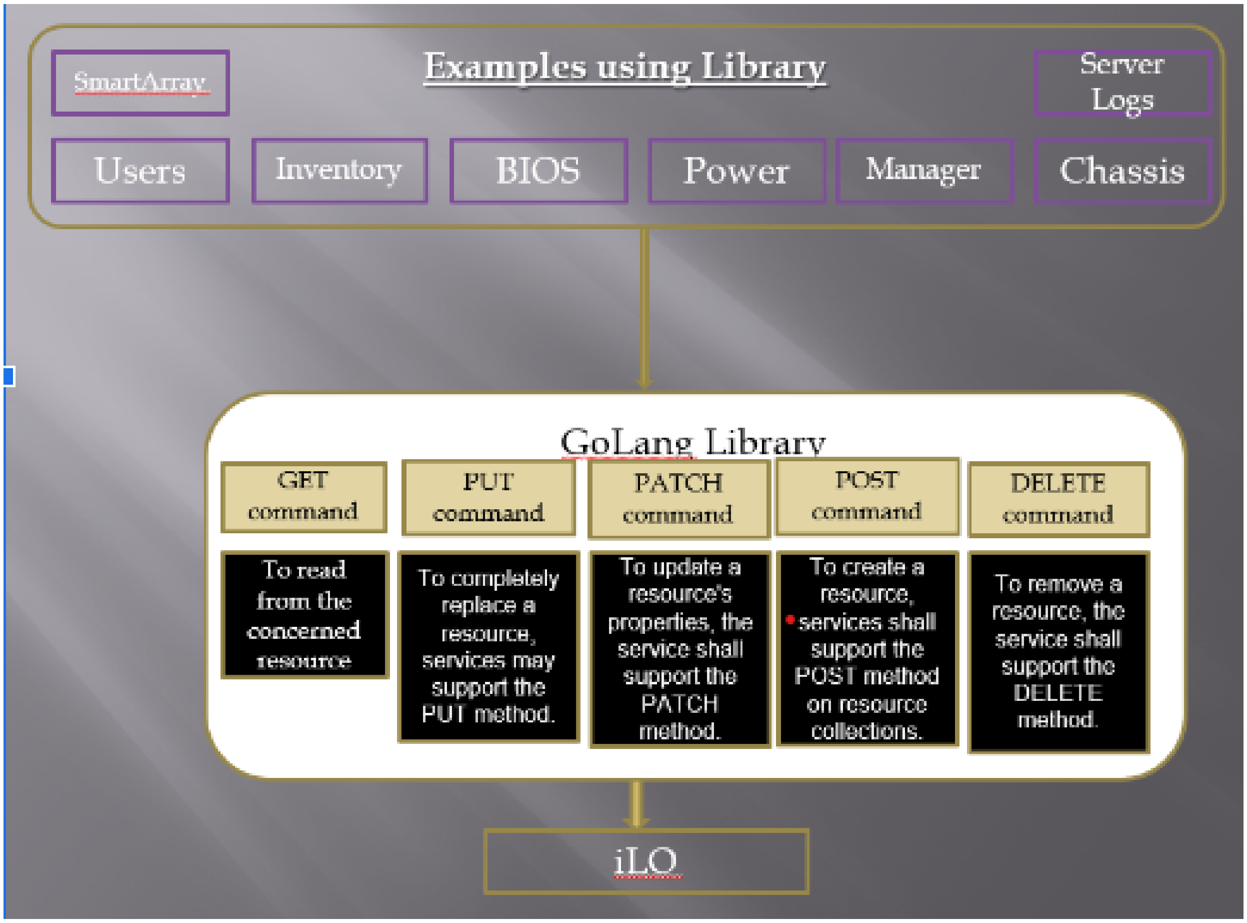
Users: HPE clients or examples like Manager, Power, Smart Array, Chassis, etc.

References

- *https://github.com/HewlettPackard/python-ilorest-library/*
- *https://github.com/HewlettPackard/python-ilorest-library/tree/master/examples/Redfish/*
- *https://hewlettpackard.github.io/ilo-rest-api-docs/ilo5/*·
- *https://www.dmtf.org/standards/redfish*

Functional Specification

Summary



To read from the concerned resource

To completely replace a resource, services may support the PUT method.

To update a resource's properties, the service shall support the PATCH method.

To create a resource, services shall support the POST method on resource collections.

To remove a resource, the service shall support the DELETE method.

System diagram for GoLang Library

Requirements Details

<Requirements should consist of inputs, process (validity checks for inputs, sequence of operations, business logic, etc.) and output. There can be explicit or derived functional requirements. This can be called out to maintain traceability. It should call out any positive and negative scenarios>

ID	***Description***	***Category***
<Req ID>	*Restful APIs*	*<Explicit, Implicit, Derived, Engineering, etc..>*

Assumptions

- *As we don't have access to HPE servers for now to test our code, so as of now we are making use of a random data generator website to test our code for REST operations which we have written.*

Caveats

- *In order to make use of go-workspace which contains the main library, first of all one needs to change the GOPATH and alter the GOROOT from the terminal to make use of this project's features.*

Platforms supported

- Windows
- Linux
- Mac

Not in Scope

<Capture any aspects not in scope of the feature specification>

Customer Experience

ID	***Description***	***Category***
<Req ID>	*< Outline how users will interact with this – graphically, via API or exposed functionality. Ensure all positive and negative scenarios are covered. Refer earlier diagrams if necessary. Detailed description of requirement including source of requirement, inputs, process, output >*	*<GUI, API, Command, etc.>*

APIs

- *https://hewlettpackard.github.io/ilo-rest-api-docs/ilo5/*.

<Add any API specifications that are needed for the customer experience>

Customer Visible Behavior Changes

<This section is mandatory and covers any changes for the feature or change request that can are visible to customers directly or via other applications that depend on this feature or change request>

External Dependencies

ID	***Description***	***Category***
<Req ID>	*< If interactions need to happen with components owned by extended stakeholders, outline use cases and expectations clearly here, so that it can be taken up by them. Refer earlier diagrams if necessary. Detailed description of requirement including source of requirement, inputs, process, output>*	*<External stakeholder>*

<Ensure that all dependent HPE applications that are affected by this feature or change request is listed here. These include other components, drivers, on-premise applications (OneView, iLOAmplifier, SUM/SUT. SaaS Applications like InfoSight, Compute Central, or published APIs from Project Martini. Please add other dependencies as applicable)>

Special Characteristics

<This section may be optional for CRs and can be marked as Not Applicable>

ID	***Description***	***Category***
<Req ID>	*< Legal, Safety or compliance requirements for the use case fall under this category if applicable. Refer earlier diagrams if necessary. Detailed description of requirement including source of requirement, inputs, process, output>*	*<GDPR, Privacy, FIPS, etc.>*

Performance

<This section may be optional for CRs and can be marked as Not Applicable>

ID	***Description***	***Category***
<Req ID>	*< Capture responsiveness, throughput, reliability, portability related characteristics with specifics. Refer earlier diagrams if necessary. Detailed description of requirement including source of requirement, inputs, process, output>*	*<Responsiveness, Throughput, Up-time, portability, etc.>*

Security

<This section may be optional for CRs and can be marked as Not Applicable>

ID	***Description***	***Category***
<Req ID>	*< Information Security of data stored or shared (Confidentiality, Integrity, Availability), Protection of Data stored within, Protection of Data shared via interfaces, restrictions of access to interfaces, mis-use cases. Refer earlier diagrams if necessary. Detailed description of requirement including source of requirement, inputs, process, output. Update based on Threat modelling as part of SDLC>*	*<Access Restriction, Password Management, etc.>*

Debuggability & Supportability

<This section may be optional for CRs and can be marked as Not Applicable>

ID	***Description***	***Validation Method***
<Req ID>	*<Outline requirements around debugging the functionality and providing support for it. Ensure that it does not conflict with any security requirements in special characteristics and HPE security guidelines. Refer earlier diagrams if necessary. Detailed description of requirement including source of requirement, inputs, process, output>*	*<Acceptance Criteria>*

Traceability

PRD or CR Requirement ID	***Feature Specification Requirement ID***	***Tracking***	***Complete/Partial***
<Requirement ID/ Title from PRD or CR>	*<Requirement ID/Title from Feature Specification>*	*<Epic in JIRA or similar where this is captured. May be same epic or different epics>*	*<Capture any gap in coverage if needed. Specify how gap will be covered>*

	<Requirement ID/Title from Feature Specification>		
	<Requirement ID/Title from Feature Specification>		

Quality

<If this is a Change Request, then consider performing a lightweight FMEA for the design and capture the high risk potential failure modes for this CR and the corresponding actions to address these failures. The details are provided in the Appendix 1 section. For a full feature FMEA this can be done on the Design phase>

References

<List of Reference documents and link if available>

Appendix 1

This section contain the failures modes that need to be considered for each of the product The following is the example failure modes that need to be evaluated for iLO and can be expanded accordingly to other products.

iLO Failure Modes

This is a living section that can be updated periodically as and when a new potential failure mode is identified that needs to be considered for various features in the product, iLO in this case.)

This is a list of various potential scenarios that need to be considered while identifying the failure modes for ***design*** *of the feature/product. This is only a sample set of the scenarios and can be modified/appended. Once the following table is filled, every question with an answer "Yes" need to be assessed for the criticality based on the severity and the chance of occurrence and must have been fixed or must have a justification for not fixing that.*

SNo	***Category***	***Failure Mode Questionnaire***	***Yes/No/ NA (Not Applicable)***
1	*Boundary Conditions*		
		1. *Does the feature fail under any under-loaded conditions (eg: zero options populated, all network interfaces disabled, minimum RAM populated, etc*	
		1. *Does the feature fail under any over-loaded conditions(eg: fully populated options, all network interfaces enabled, maximum RAM populated, the system is running at high utilization etc*	
2	*Performance Challenges*		
		1. *Does the feature fail when the processing is not complete in the expected duration*	
		1. *Does the feature fail when the response is not received within the expected limit*	

3	*Synchronization challenges*		
		1. *What happens if the depending task or peer misbehaves with the synchronization primitives*	
4	*Feature behaviour in various Security modes*		
		1. *Does the feature fail in L6 and L10 within the Factory in Production or any mode*	
		1. *Does the feature fail in Factory mode*	
		1. *Does the feature fail in Production Mode*	
		1. *Does the feature fail in High Security Mode*	
		1. *Does the feature fail in FIPS mode*	
		1. *Does the feature fail in CNSA mode*	
5	*Privileges*		

		1. *Does this feature fail due to the lack of proper user privileges?*	
6	*Interoperability challenges*		
		1. *Does this feature 'rely on'/'interact with' a 3rd product and fail to interoperate due to protocol version mismatch*	
		1. *Does this feature fail to interoperate with 3rd party product due to backward compatibility of the 3rd party product/service*	
7	*Hardware Failures*		
		1. *Does this feature fail due to NAND Failure*	
		1. *Does this feature fail due to SRAM Failure*	
		1. *Does this feature fail due to Options Failure*	
		1. *Does this feature fail due to Fans Failure*	
8	*Security Crypto Algorithms*		

		1. *Does the feature fail if the supported algorithms are not sufficient to interoperate with the peer (Eg ECDSA not supported by CA)*	
		1. *Does the feature fail because of lack of sufficient entropy while generating keys*	
9	*Data Structure limits*		
		1. *Does the feature fail if input sizes are larger than the supported data structure handling the input buffer (eg: DLBUF is 32MB)*	