

SALSA APIs and client

1 RESTful API

The table below lists the APIs of SALSA framework to manage the IoT Cloud System. By default, SALSA can be deployed on a webserver or run as standalone and expose RESTful service. The endpoint for the RESTful includes the IP and port, for example as below, the full endpoint can be: <http://localhost:8080/salsa-engine/rest>.

Nr.	RESTful API: /salsa-engine/rest/				Description
	Type	Resource URL	Consumes	Produces	
1	PUT	/services/{serviceName}	MULTIPART_FORM_DATA		Submit and deploy a service*. Tosca is embedded in a HTML form.
2	PUT	/services/xml	Application/XML		Submit and deploy a service. The TOSCA is the data of the request.
3	GET	/services/{serviceId}	-	Application/XML	Get the deployment structure of the application.
4	POST	/services/{serviceId}/redploy	-	-	Redeploy the service
5	DELETE	/services/{serviceId}	-	-	Remove the service
6	POST	/services/{serviceId}/nodes/{nodeId}/instance-count/{quantity}	-	-	Deploy more units.
7	POST	/services/{serviceId}/nodes/{nodeId}/instances/{instanceId}/action_queue/{actionName}	-	-	Execute a reconfiguration. The command is put in to a queue.
8	DELETE	/services/{serviceId}/nodes/{nodeId}/instances/{instanceId}	-	-	Delete a specific instance by ID.
9	GET	/viewgenerator/cloudservice/json/compact/	-	Application/JSON	Show application structure as a tree.
10	GET	/viewgenerator/cloudservice/json/list	-	Application/JSON	Get the list of the applications.
11	POST	/elise/communication/queryUnitInstance	-	Text/plain	Start a query from external service for the configuration information.
12	GET	/elise/manager/query/{queryUUID}	-	Application/JSON	Check the status of the information collection process of (11)
13	GET	/elise/unitinstance/{instanceUUID}	-	Application/JSON	Get the more information of an instance after (12). The instanceUUID is get from (3) or (9).

* The term "cloud service" above refers to the whole IoT topology or cloud services in a single description.

The usual flow of using API is: Using (1) or (2) to submit the TOSCA and deploy the application. When the application is deployed, use (11) to collect more information and (12) to query this information. Due to the process of deployment and collection may take time, (3) and (12) can be used to check this progress.

2 Salsa-client

Sys-admin can use salsa client to manage the system. User can see help for the details of available commands:

```
./salsa-client --help
```

```
Usage: java -jar salsa-client.jar
```

```
SALSA Java command-line client
```

```
-a (--address) <address> : The address of the salsa engine (default:
                           128.130.172.216)
-h (--help)                : Print the help and exit (default: true)
-p (--port) <port>         : The port of the salsa engine (default: 8080)
```

Type 'java -jar salsa-client.jar help <command>' for help on a specific command.

Available commands:

```
conductor-list-collector: Get the list of available collector plugins.
conductor-push-collector: Add a collector plugin to conductor to collect information.
conductor-start         : Start a conductor to collect the information
conductor-stop          : Stop a conductor by ID or at salsa-engine
instance-deploy         : Deploy one or more instances of a service unit.
instance-info-collect   : Ask collector to start gathering the information. The process may take time.
instance-query          : Get the information of an instance by ID.
instance-remove         : Remove an running instance.
meta                    : Get metadata of the SALSA which is connected.
service-list            : List the current managed cloud services
service-remove          : Undeploy all components and remove the cloud services.
service-status          : Get all the list of instances
service-submit          : Submit a TOSCA to start a deployment.
syn                     : Send message the synchronize pioneers.
```

Beside the commands that are similar to APIs in Section 1, there are command lines for distributing the collectors and integrate with other services. Conductor is in charge of managing collector module and communicates with SALSA. The salsa-client can send the request to salsa-engine or salsa-pioneer to start/stop a conductor. When a conductor is running, salsa-client push/configure collector modules on the conductor.