**Swordfish Basic Web Client – User Guide**

|  |  |  |
| --- | --- | --- |
| **Author** | **Date** | **Version** |
| **Sravanthi** | **25/04/2018** | **0.1** |
|  | **6/6/18** | **0.2** |
|  | **07/06/2018** | **0.2.1** |

Contents

[1 Overview 2](#_Toc516135241)

[2 Introduction 2](#_Toc516135242)

[3 Environment 2](#_Toc516135243)

[3.1 Default Port 3](#_Toc516135244)

[3.2 Default Web server 3](#_Toc516135245)

[4 Swordfish Basic Web Client Functional Areas 3](#_Toc516135246)

[4.1 Basic Web Client Application Login 3](#_Toc516135247)

[4.2 Logout 4](#_Toc516135248)

[4.3 Add Swordfish Services 4](#_Toc516135249)

[4.4 Delete Swordfish Service 9](#_Toc516135250)

[4.5 Add Collection 10](#_Toc516135251)

[4.6 Delete Collection 11](#_Toc516135252)

[4.7 Edit Properties 11](#_Toc516135253)

# Overview

SNIATM Swordfish has been designed around management use cases that focus on what IT administrators need to do with storage equipment and storage services in a data centre. As a result, the API provides functionality that simplifies the way storage can be allocated, monitored, and managed.

# Introduction

The Swordfish Basic Web Client gives a clear demonstration of collections and its sub collections data in a blade interface.

The detailed view of a real time system’s data is organized in a hierarchical manner showing the kind of relation (members, links, properties, etc.) between them using a blade interface design approach.

The Swordfish Basic Web Client can connect to one or more Swordfish (or Redfish) services (including emulators), and present in a web UI frame the entire system hierarchy. The basic web client also provides basic capabilities to modify configurable Properties (as specified in the schema), as well as providing a basic interface to add or remove elements from the service through a Web UI.

# Environment

Node.js and npm are essential to run the Swordfish Basic Web Client. Node powers client development and build tools. The npm package manager, itself a node application, installs JavaScript libraries.

The node.js environment installs and runs a web server on the system. To run the client, you will first need to launch the web server, then point a web browser at the web server to run the Swordfish Basic Web Client interface.

Note: you may also need to open appropriate ports on your system’s firewall to enable appropriate operation of the Swordfish Basic Web Client.

|  |  |
| --- | --- |
| **Operating System** | Windows/Linux |
| **Node.js** | v.8.1.1 |
| **Npm** | v5.0.3 |
| **Angular/cli** | 1.4.5 |
| **Express** | 4.15.5 |
| **IDE** | WebStorm (optional) |

## Default Port

Node.js runs on 3000 port by default

\*\* For configuring port options, please refer SNIA Developer guide.docx

## Default Web server

Using http-server a local web server is created which runs the application locally. There are no additional steps required to install http-server as the versions which need to install was specified in the package.json files.

\*\* For installation instructions please refer to SNIA Installation Steps.docx

# Swordfish Basic Web Client Functional Areas

The Swordfish Basic Web Client is composed of the following functional areas to access the data from Swordfish systems.

## Basic Web Client Application Login

Allows the User to avail the features of Swordfish Basic Web Client by authenticating.

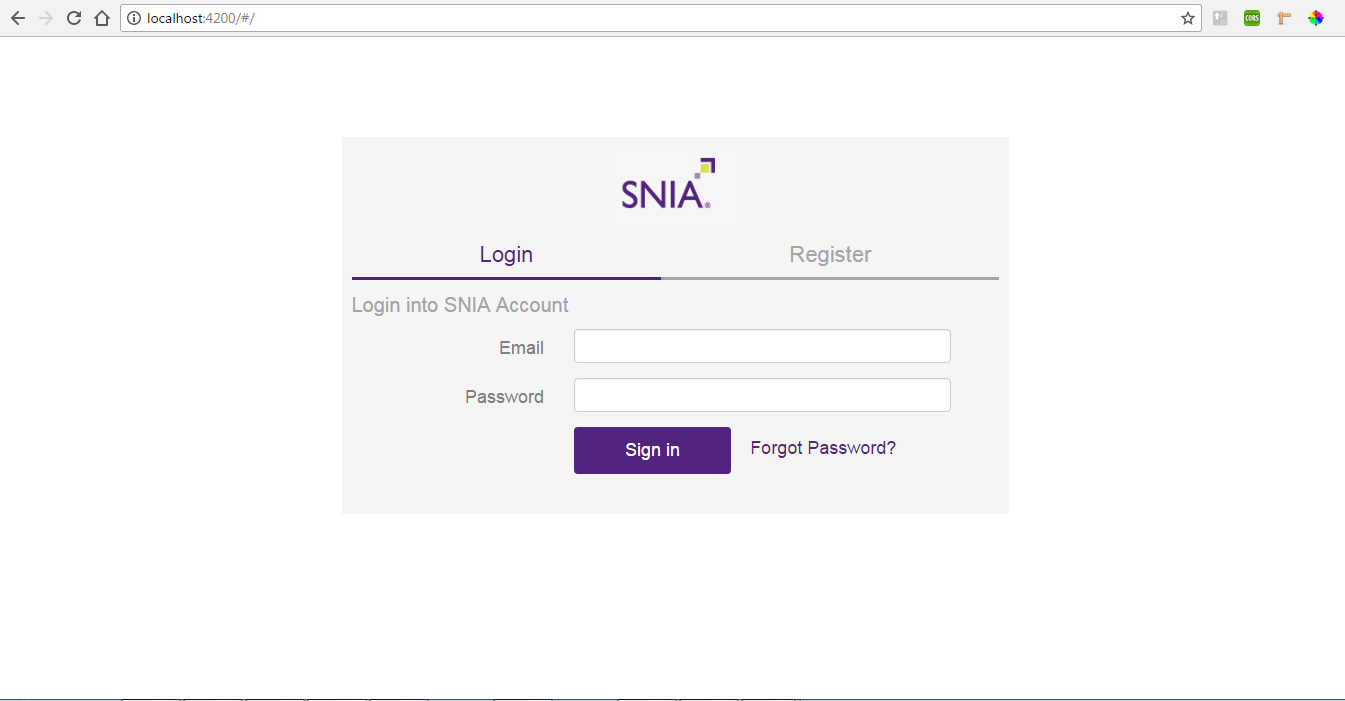
User needs to provide default Username and Password.

Email: [swordfish@snia.org](mailto:swordfish@snia.org)

Password: snia@123

This credential can be changed.

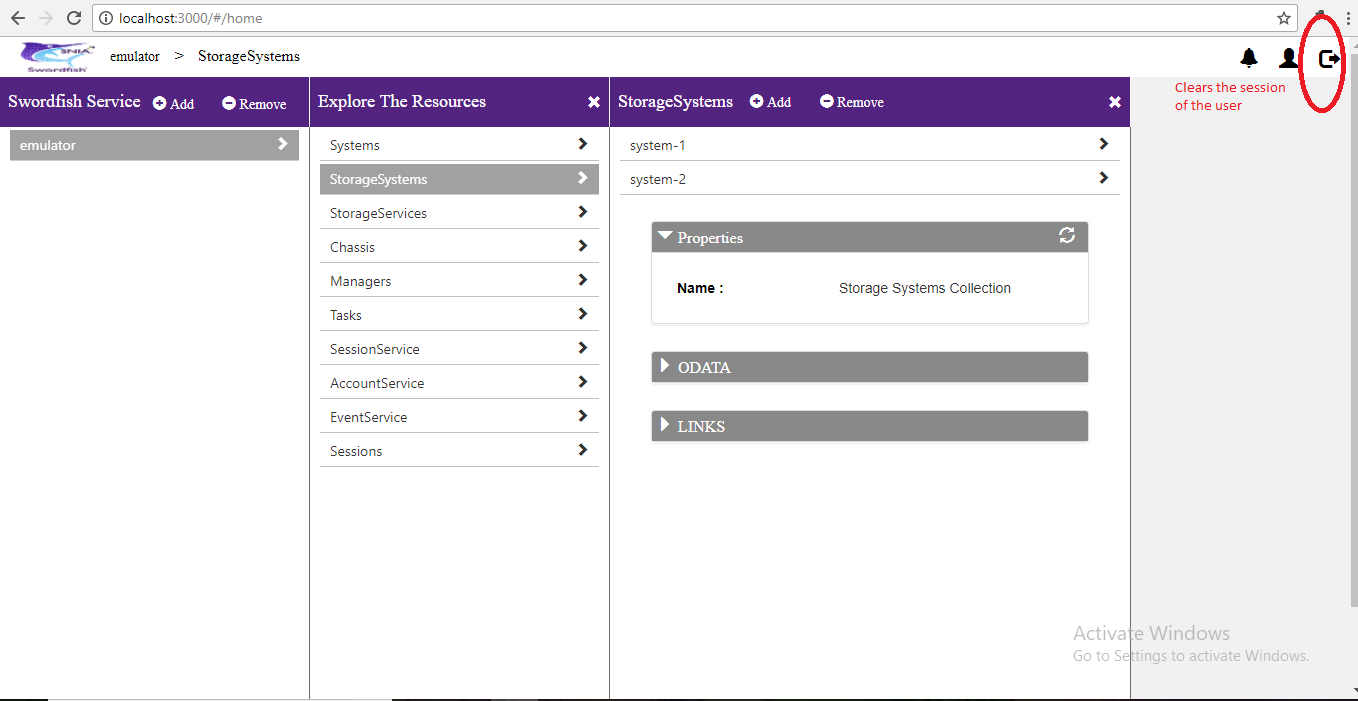
\*\*Please refer to the Swordfish Basic Web Client Developer guide for instructions on how to change the default Email and password



1. Enter Email and password in the input fields and click on Sign In button

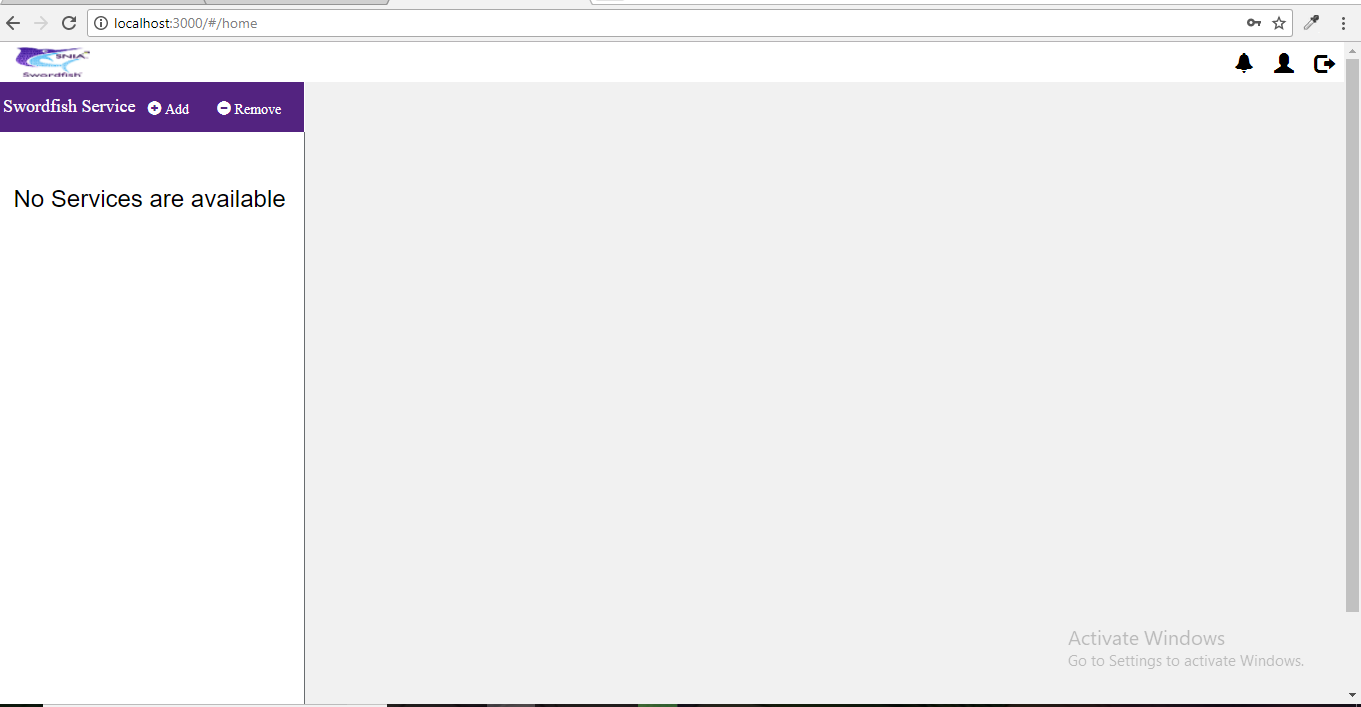
## Logout

Deletes the Session and Services added by the user. A service call is made to the respective API Endpoint to delete the Session created to prevent DOS attacks



## Add Swordfish Services

* A list of Services is shown in the blade under Swordfish Services if they were added already
* A message showing “**No Services are available**” is shown if there are no Services added

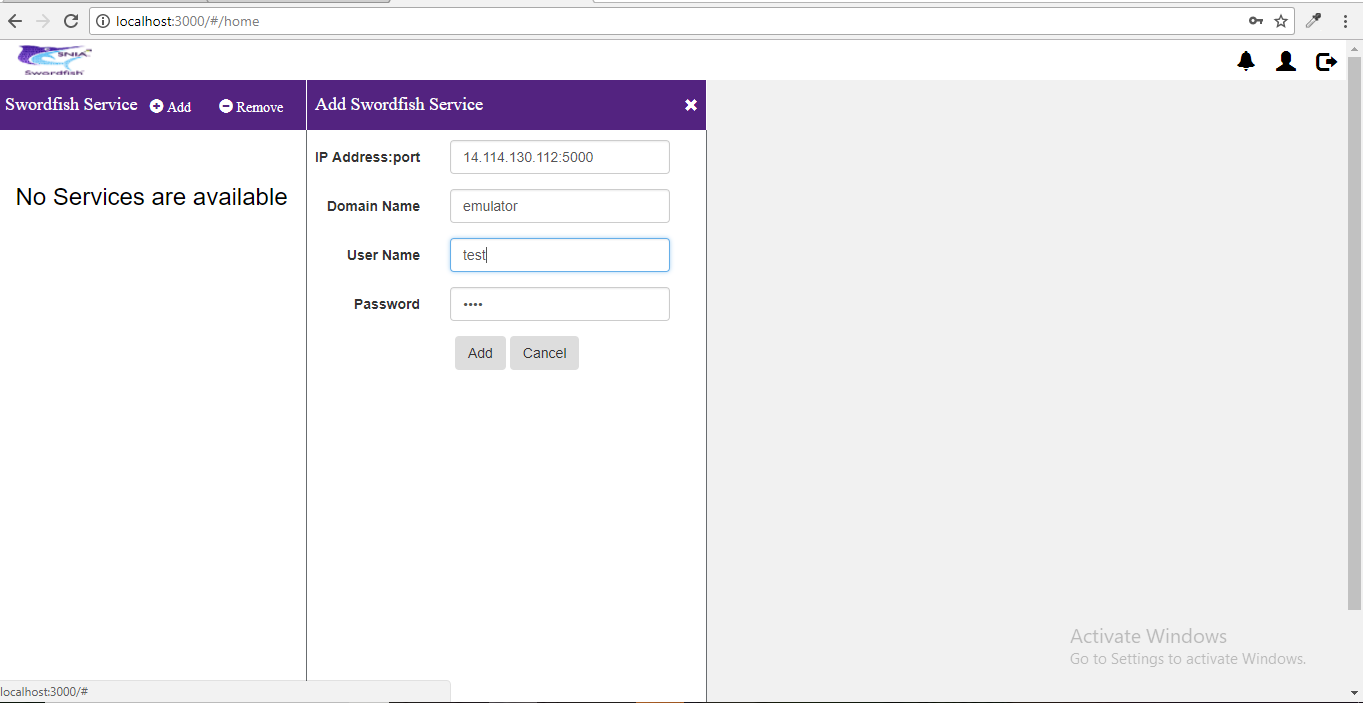


* Swordfish Basic Web Client allows user to Add/Remove services to the system
* On Add of a new Service, a new blade will visible to user with a form containing IP address, Domain Name, User Name and Password.
* User Name and Password are used by the server for authentication purpose.
* **IP Address**: IP Address of server where real time systems data resides. Port number need to be included while adding the IPaddress

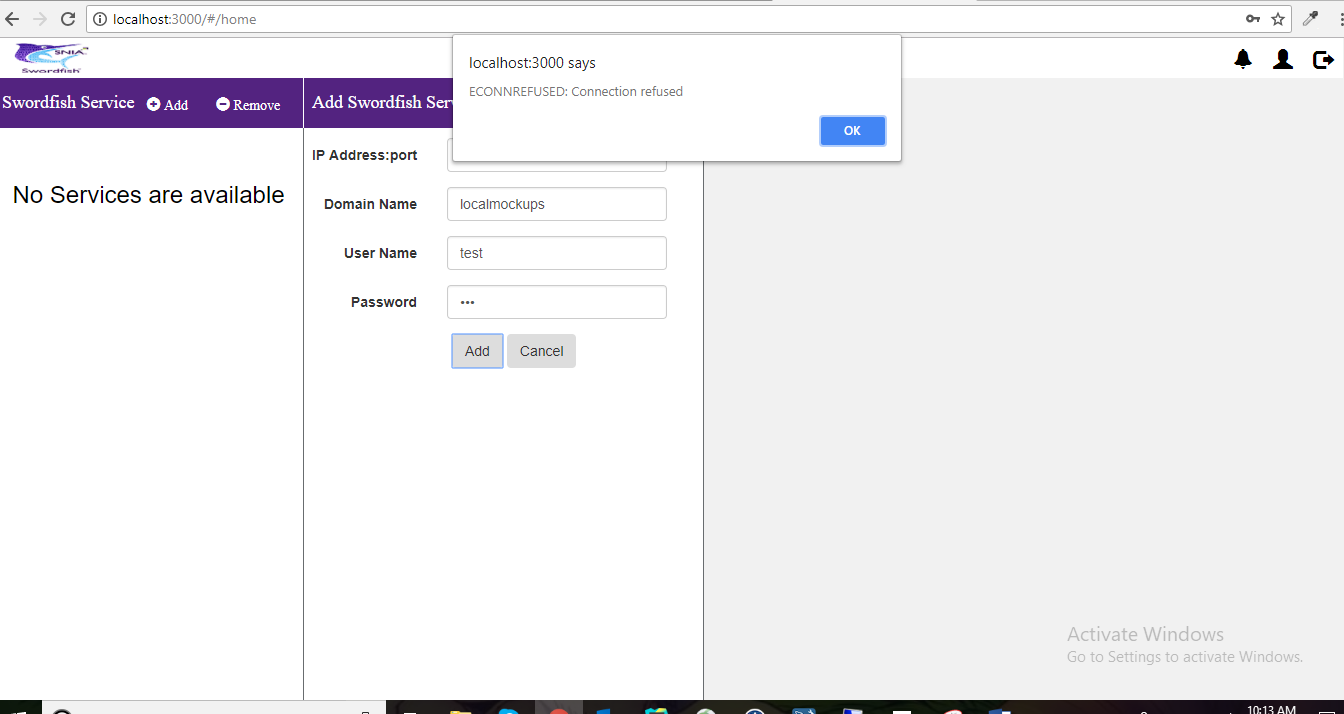
**ex**: http://localhost:8089

**ex: 10**.12.14.15:9000

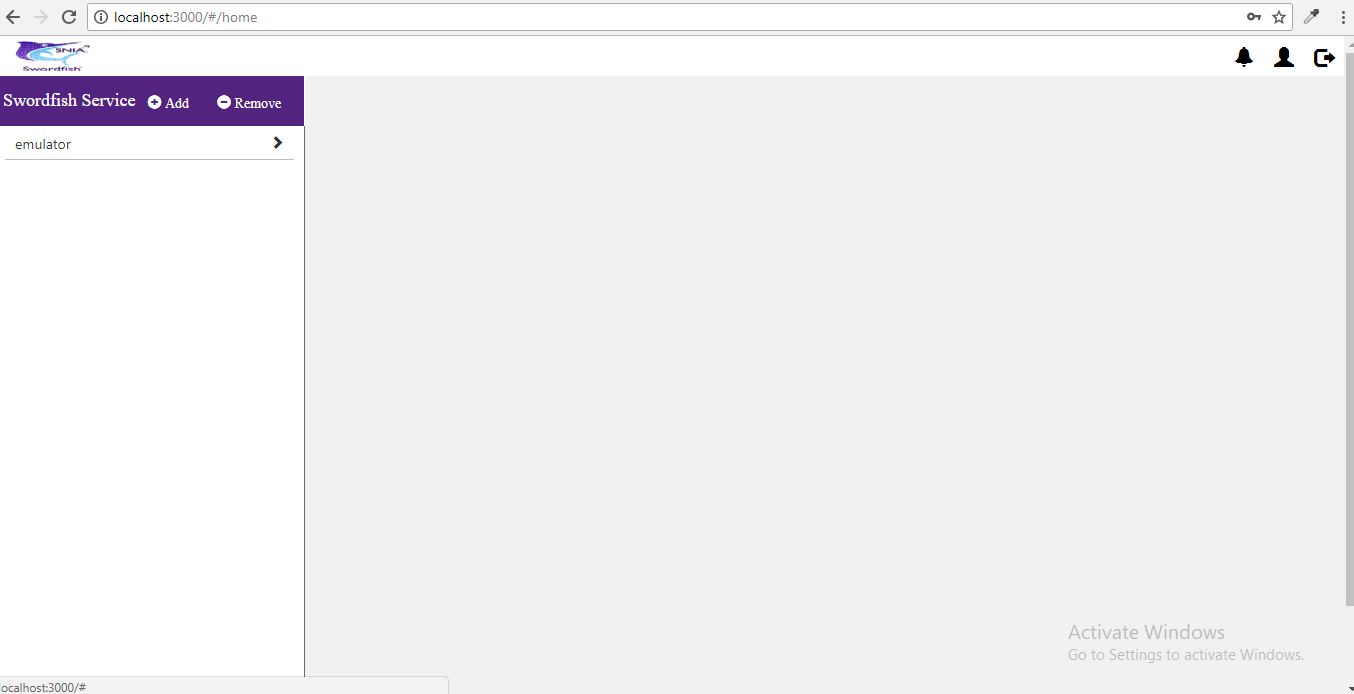
* IP address Port Number
* **Domain Name**: Identification string (For user understandable IP Address are mapped to domain name)



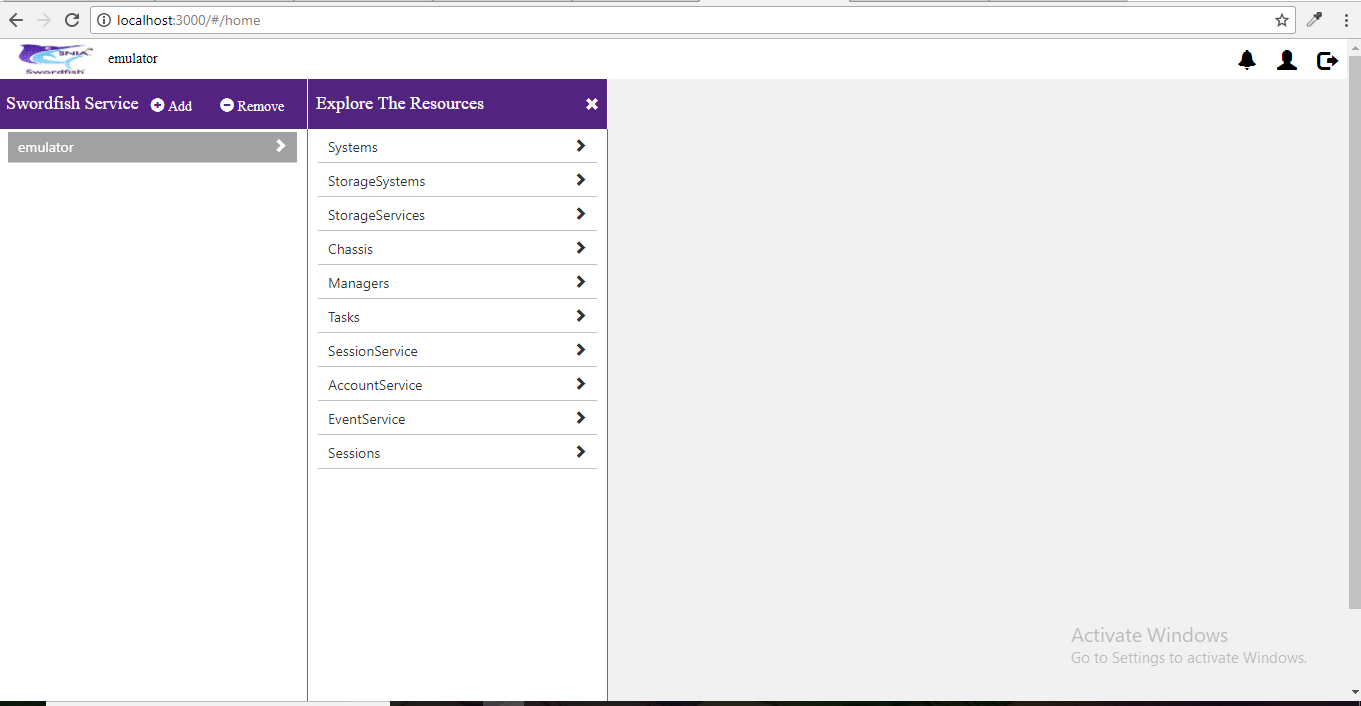
Any errors while adding the services are handled and shown to the user in the alert box indicating the cause



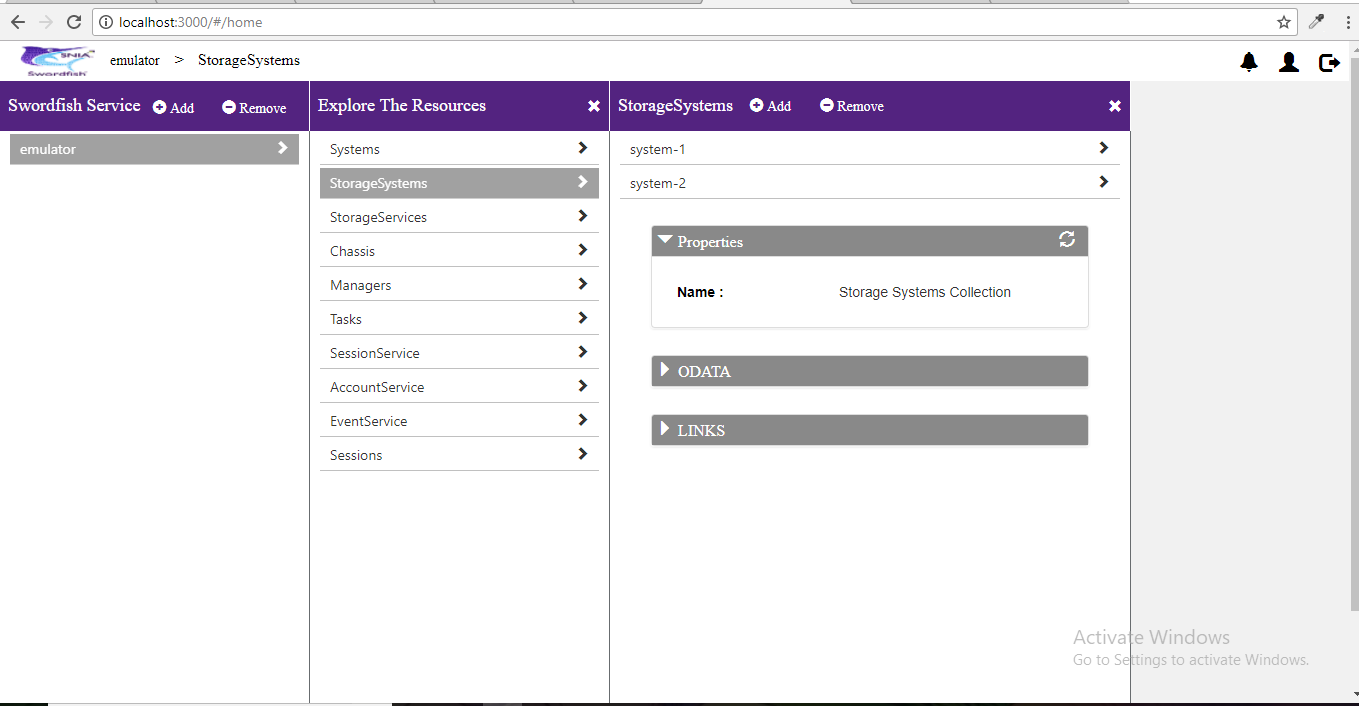
* Else, service is added to the list as shown below

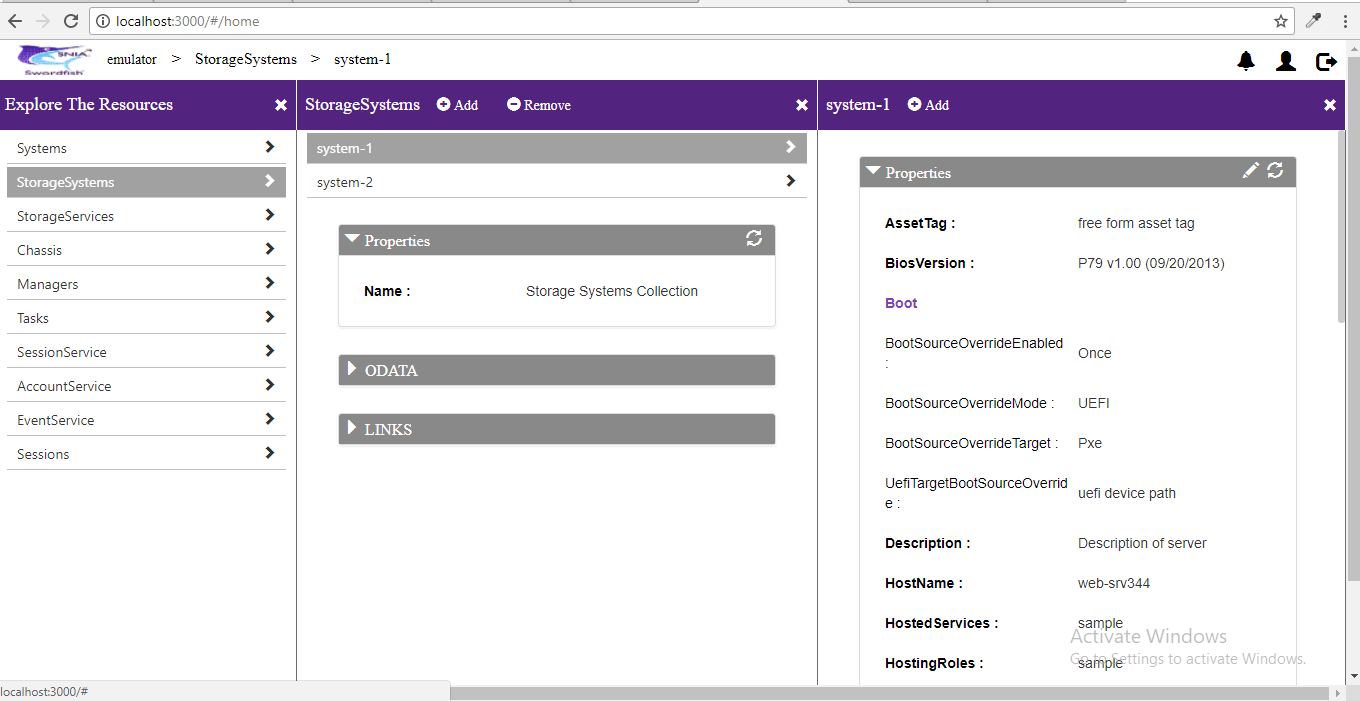


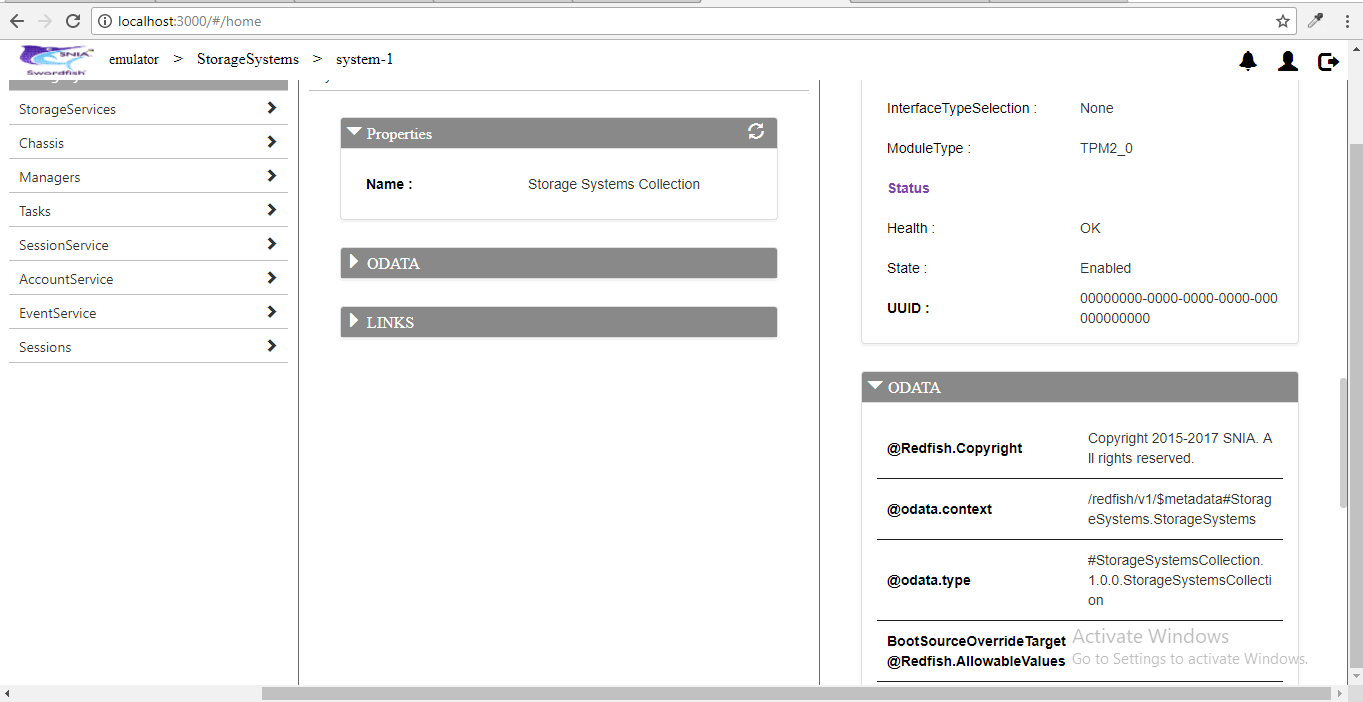
* On click on the newly added service user can be able to see information related to that service and the collections present under it



* User can explore through the collections

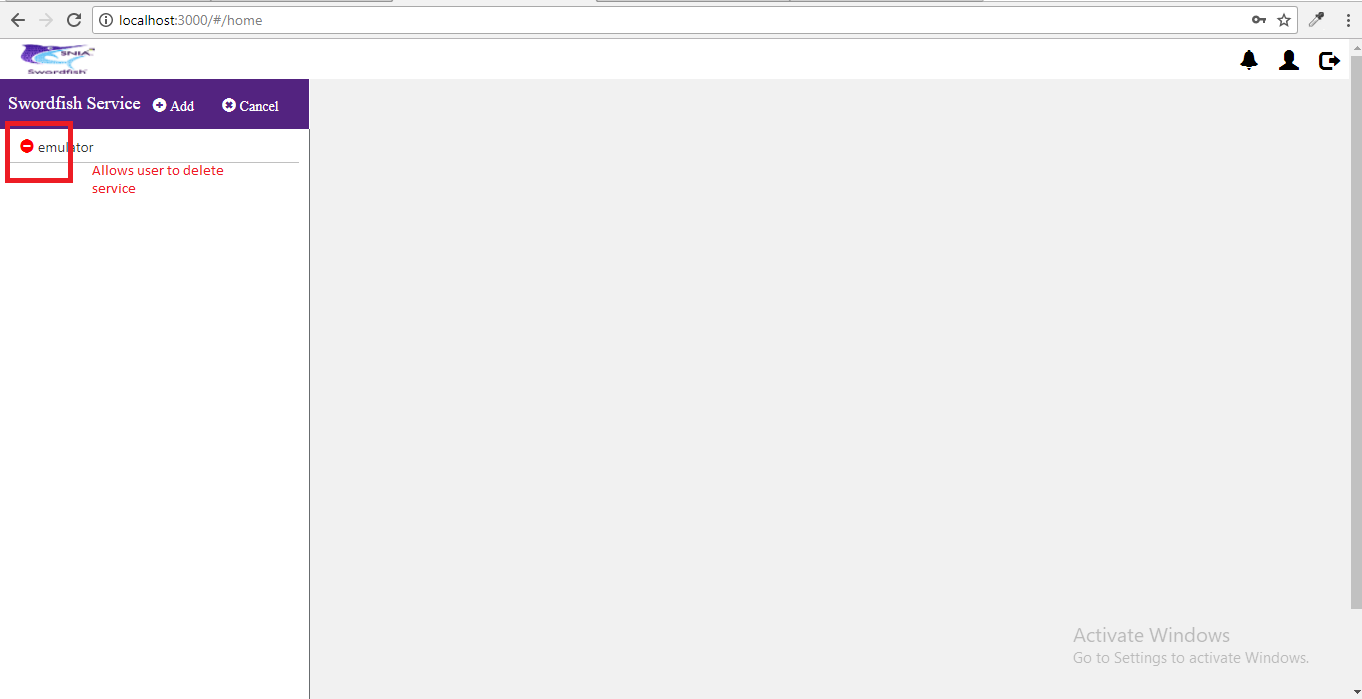




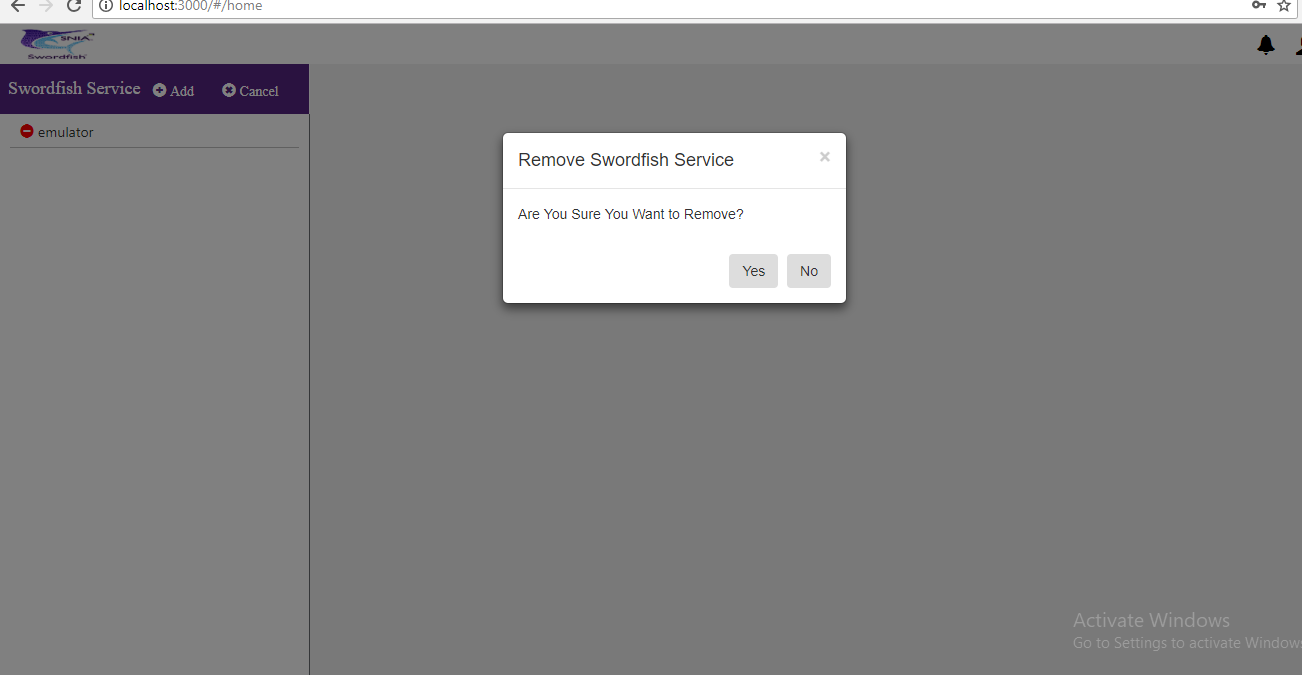


## Delete Swordfish Service

Deletes the added Swordfish Services on user approval. A service call is made to the respective API Endpoint to delete the session created to prevent DOS attacks



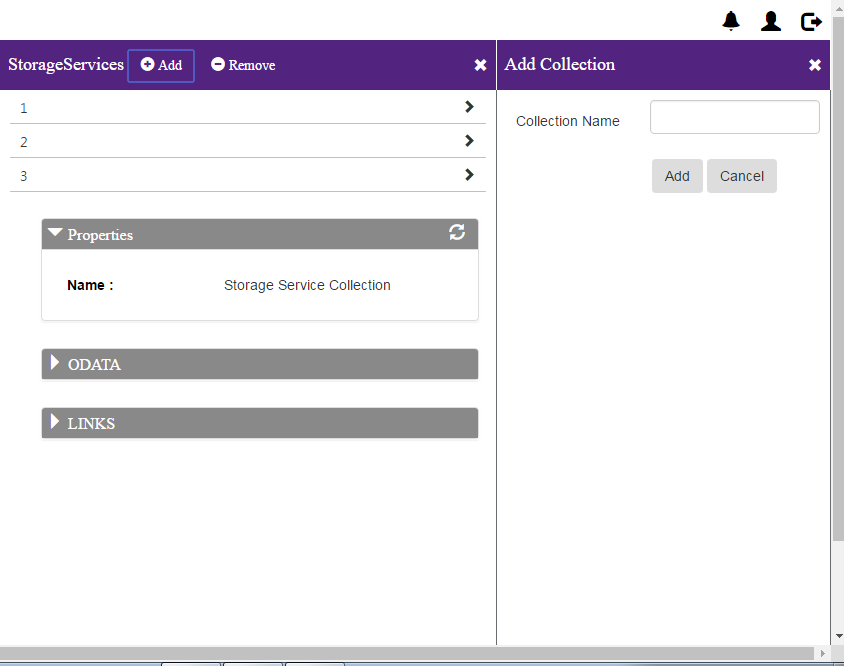
* User is prompted with an alert to delete the selected Service



* If user wish to delete, click on Yes button. Else click No
* On Yes, the session ID created will be cleared
* On No, user will be back to original state, able to see the added service and can perform regular actions

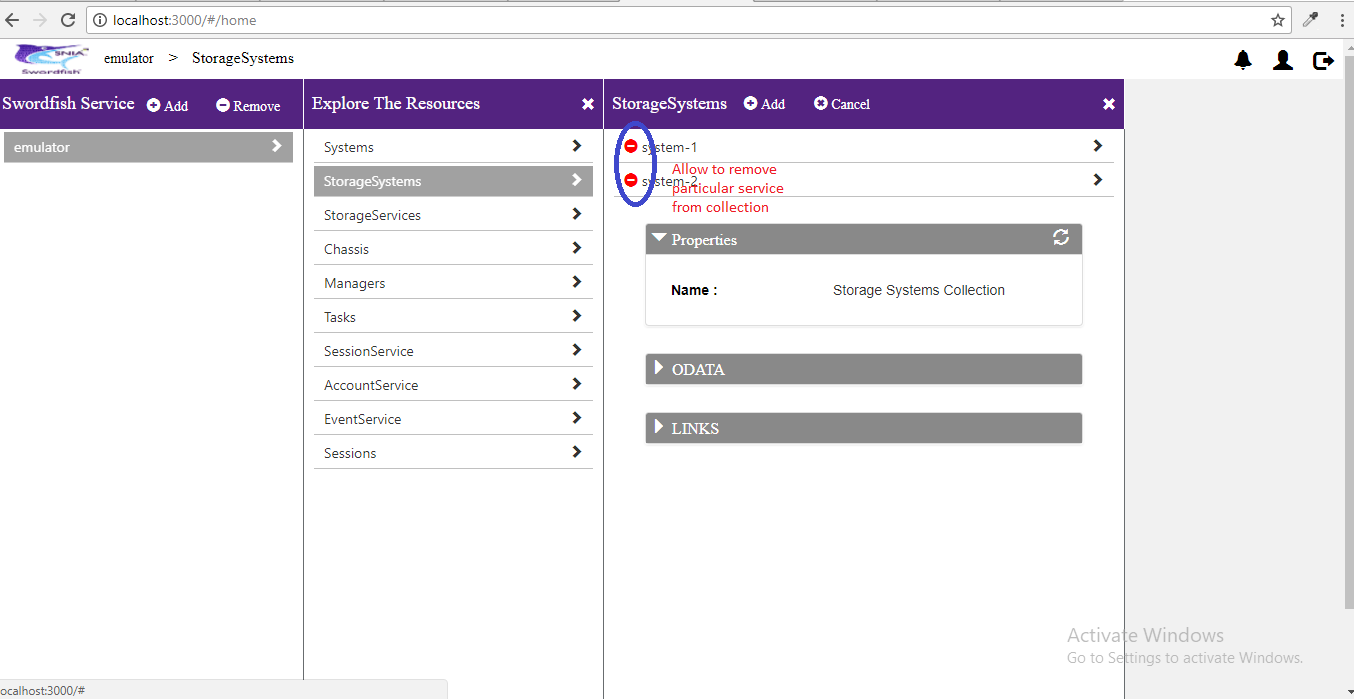
## Add Collection

An ‘add’ option is provided at collection level to add a new collection to the existing



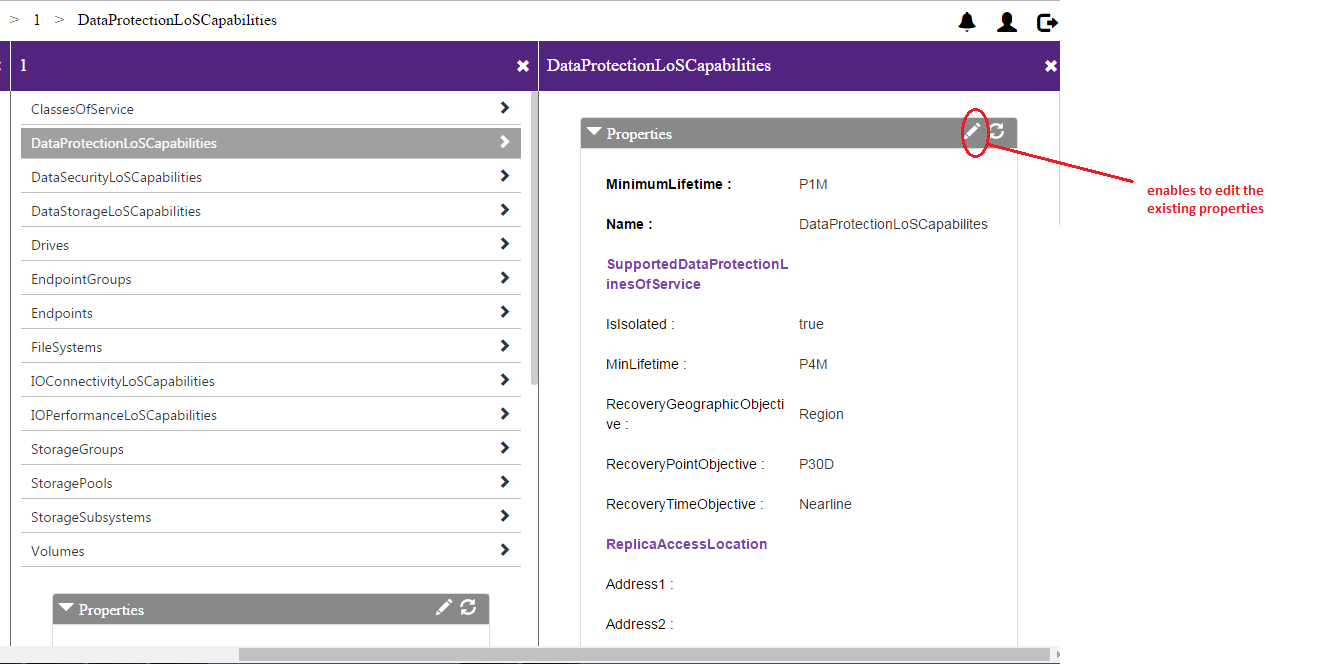
## Delete Collection

An ‘remove’ option is provided at collection level to remove existing collection.

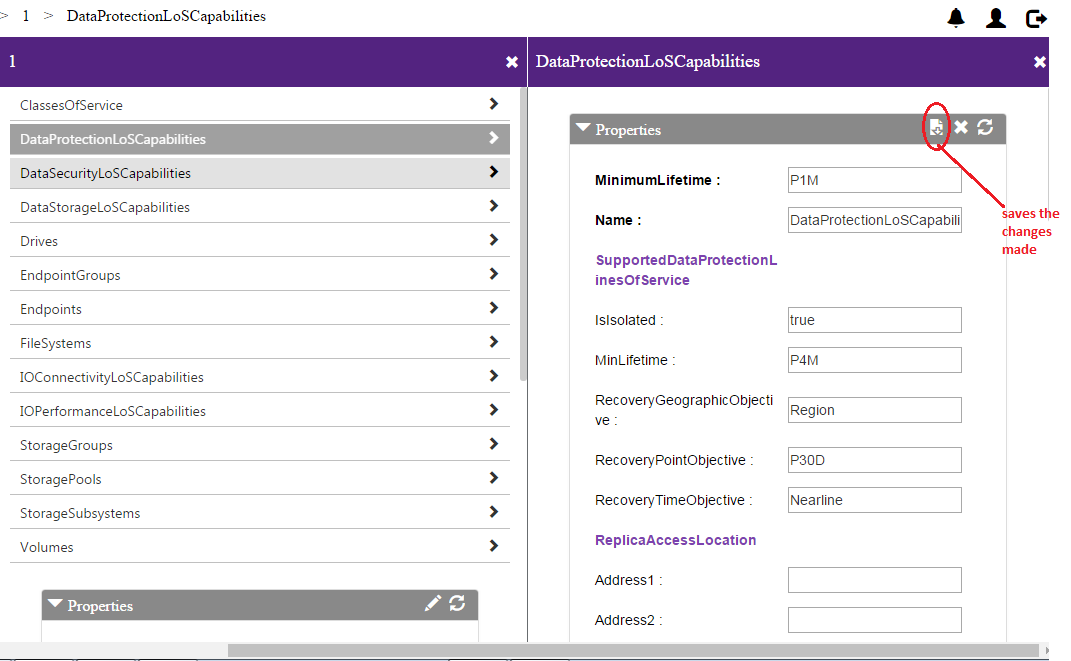


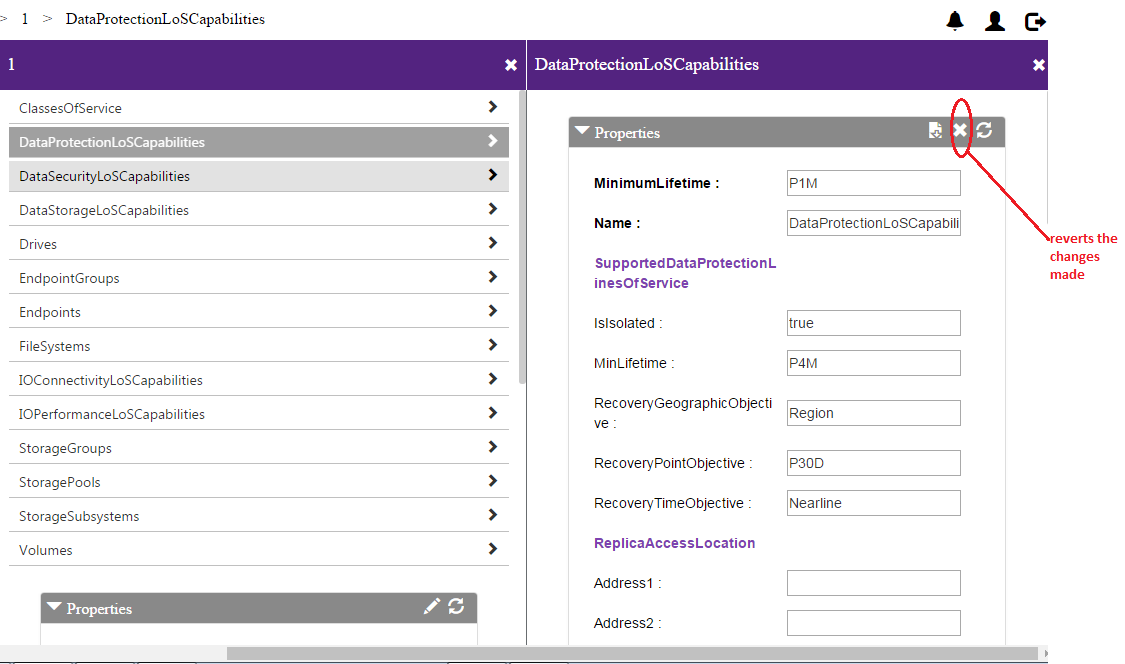
## Edit Properties

* A series of options are provided to edit, save or cancel the changes made to the existing properties.
* User can click on the pencil icon on the properties tab to edit the existing properties



* On Edit, user can save the changes or cancel them





* Without refreshing the entire page, a refresh option is provided at the blade level to view the immediate changes made

