Auto-generated Isilon OneFS Management Interface

Sponsor: Dell EMC Isilon
Mentors: Grady Karp, Kelsey Bueno
Matthew Cox, Matthew Ferran, Samuel Smith, Jacob Shelton, Jerome Stylos

Background

Dell EMC Isilon provides a scale-out storage solution using a series of servers that are strung together into clusters, running an in-house operating system called OneFS. These clusters can be configured by clients through a web interface that is transparently powered by a RESTful API. The current interface is tightly coupled with the OneFS release cycle, and requires significant changes to support new API endpoints. This project’s goal is to leverage the API’s self-describing nature to dynamically generate a management interface for the client’s OneFS version on the fly, reducing the need for maintenance as new API endpoints will be supported automatically.

Goals

- Generate a management interface for any and all self-described versions of the OneFS API
- Allow all endpoint contents to be viewed alongside their corresponding descriptions, as well as edited if supported by the endpoint
- Support authentication for any valid user account
- Implement a general workflow that goes beyond single endpoint viewing and editing
- The final product should be usable and intuitive for a user who has no prior knowledge of OneFS

Architecture

Our system is decomposed into three major components: the endpoint fetcher, the endpoint processor, and the user interface. These components are pipelined as shown here.

(endpoint identifier)
- i.e. /3/cluster/config
  (API location)
  i.e. localhost:8080/api
  (user credentials)

Endpoint Fetcher
  (raw endpoint contents)

Endpoint Processor
  (raw endpoint self-description)
  (parse-tree structured according to endpoint self-description, aggregating endpoint contents, input schema, and output schema)

User Interface

Future Work

- Add filtering methods for endpoints with long contents
- Revise endpoint self-descriptions for certain endpoints whose capabilities are not well represented
- Add support for input validation according to the current endpoint’s supported inputs
- Write more rigorous tests for our system’s various components

Glossary

- Cluster: A group of computers working together as one
- API: Application Programming Interface
- REST: Répresentational State Transfer

Technologies

- JS
- git
- React
- Redux
- Jenkins
- Bootstrap
- vmware
- npm
- Heroku
- Hardware

Acknowledgements

We would like to thank our mentors Grady Karp and Kelsey Bueno, Isilon’s user experience team, our professor Aaron Crandall, and our usability study participants for their help in leading this project to completion.