

IT Solutions: Discovery Session Exporter

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

David Martin Salzberg
Spring, 2023

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Rosanne Vrugtman, Department of Computer Science

IT Solutions: Discovery Session Exporter

CS 4991 Capstone Report, 2022
Martin Salzberg
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
dms4ad@virginia.edu

ABSTRACT

ScienceLogic, a Reston, VA-based IT management software developer company, decided that manually copying a discovery session, a record of a device's connection to one of their SL1 systems, from one SL1 instance to another was too cumbersome and time-consuming. The solution involved the automation of curl "post" and "get" requests to the ScienceLogic SL1 API via VPN. I utilized the Python "requests" library to import a discovery session from a SL1 instance to the local system in .json format. I then exported that session to the target SL1 instance. Any dynamic fields were updated locally before I exported the session to the target system. We completed the project on schedule. The tool may have future use in saving lost time filling in every field as well as eliminating the possibility of human error while doing so. Moreover, the tool may see usage during testing because of its ability to replicate a difficult-to-solve error from one SL1 instance to another. The SL1 API has had difficulty with exporting credentials attached to a discovery session, some of which have encrypted fields that do not encrypt on the target system in some versions of the SL1, so these systems will need to be updated if that issue is to be avoided while using the tool.

1. INTRODUCTION

How long does it take to fill out a form that needs a name, address, contact information, emergency contacts, social security number, driver's license number, insurance information,

and more? What if that process took three times as long as it already does? ScienceLogic's back-end developers and software testers perform this task multiple times a day with discovery sessions on their SL1 systems, consuming hours a day per person.

The SL1, which is ScienceLogic's software used by IT professionals, contains several useful tools to help manage their workload. Discovery sessions, which are records of devices' connections to a SL1, include the following fields: name, ID, URL, date/time, credentials (often multiple), templates, and more. Some of these fields have their own sub-fields, such as credentials, which require a name, ID, URL, date/time, credential type, and credential service (of which there are hundreds). Among these fields, a handful are considered "dynamic," meaning that the field would change from system to system. As we continue to go deeper into filling out the data for a discovery session, it becomes apparent that this process can be cumbersome and tedious, especially for those who perform it multiple times a day, not to mention the possibility of losing data or filling it out incorrectly, given the large amount of data that a human has to manually parse on the source SL1 and type out on the target SL1.

When ScienceLogic was in its early stages as a small company, this process was not as much of a problem, but as the company grew to over 700 employees, many more people were carrying it out, adding up to countless company hours taken up by a tedious task that was quickly becoming a burden. To avoid the trouble that

this growing issue would inevitably cause, I was tasked with designing a tool that would automate the process, drastically reducing the time consumed as well as minimizing the possibility of human error.

2. RELATED WORKS

A similar case of a beneficial switch from manual work to automation is that of HSBC, a banking organization. HSBC partnered with the CloudBees software platform in 2015 to automate many internal mundane tasks, saving much of their time that was previously lost. With the automation provided by CloudBees, HSBC can now work more efficiently to “get new digital products and services into [their] customers’ hands more quickly and securely” said chief technology officer of HSBC Ian Haynes via Flinders (2021). This case is even more applicable following the shift to more computing-oriented lifestyles because of the COVID-19 pandemic, with “customers [shifting] more and more of their banking online.” – Haynes, Flinders (2021). As a result, HSBC has become a more efficient organization on the administrative level, sales, and now even their customer service with the LivePerson AI platform launched in 2019, comparable to how my project’s implementation of automation will save hours of work in a matter of seconds for back-end

employees, making it more efficient for ScienceLogic to tend to their customers on the front-end.

3. PROJECT DESIGN

The discovery session exporter involves multiple steps during usage. In short, it imports a session and related sub-fields from a source SL1 to the local system, then exports that data to the desired target SL1.

3.1 Review of Discovery Sessions

Discovery sessions are part of the SL1 interface, and each contains an extensive list of fields that must be filled out. Some of these fields are considered “dynamic,” which indicates that the contents of the field would change from system to system, such as the edit date. Additionally, in some cases a field does not exist on the target system, in which case it would have to be replicated separately from the discovery session. The creation of these fields was a critical part of the project because it determined whether the target system would approve a discovery session to be imported.

Figure 1 shows the interface used to create a discovery session, and displays each of the fields required to create one. Some of the fields are optional, like description or any of the Detection and Scanning fields.

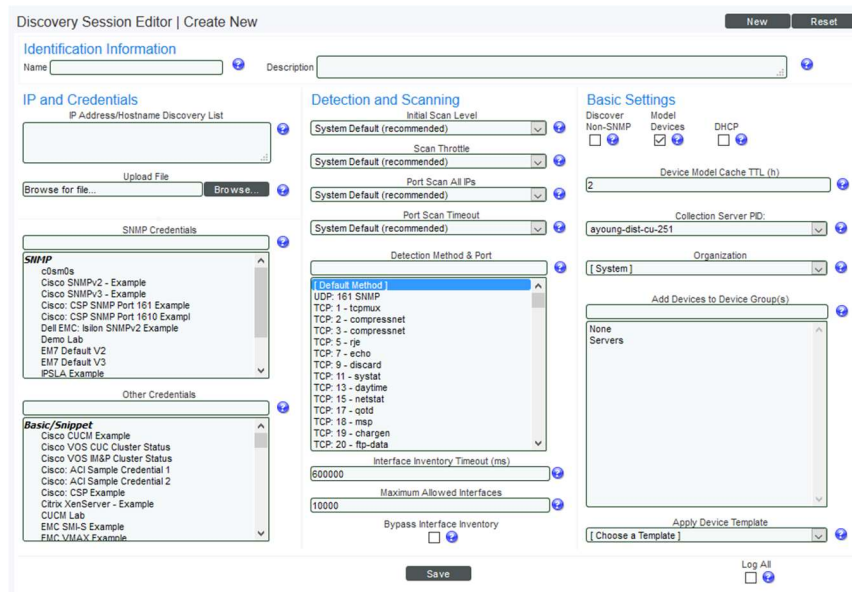


Figure 1: Discovery Session Creation Interface

To export a discovery session from one SL1 to another, the session must first be exported to the local system. This was done by writing a program using the Python requests library, which allowed me to utilize the SL1 API. The API was accessed via a VPN connected to my computer, which then let me use a SL1's IP address to view its contents with a web browser. Each SL1 had its own API which had a section that displayed each discovery session associated with it, and the details that pertain to each discovery session, which are its fields.

By analyzing multiple SL1 APIs, I was able to coordinate my program in a versatile way that could access any system as long as the user of the program had its login credentials. The most critical parts of developing this tool, as mentioned above, were the fields that needed to be replicated from the source SL1 to the target SL1 because an export of a session would fail if one of those fields did not exist by the time the session was created on the target system. The fields that fall under this category are credentials, organizations, and templates.

3.2 Exporting a Discovery Session

To begin the operation, we first imported a discovery session to the local system as a .json file, which allowed for ease of parsing and organization because it automatically separates each item into a list format, which is much more manageable than manually parsing a continuous string of words when writing a program. Next, the discovery file was parsed to locate its associated credentials, which were then searched for on the source SL1 API's credential section using search filters that were available as part of the API's features. Once the correct credentials were found, they were also exported as a .json file to the local system. This process was repeated for organizations and templates.

After all files were on the local system, we then created new credentials, organizations, and templates on the target system using the

information that was now contained within the files we imported. Following this step, we created a new discovery session on the target SL1 and searched the target SL1 for the newly created credentials, organizations, and templates which we then linked to our new discovery session. After all the data was linked, we published the new discovery session on the target SL1.

4. RESULTS

The project was completed on schedule and is now in use by back-end employees, especially those who work in software testing. With the use of this tool, ScienceLogic employees are now able to copy discovery sessions between systems much more efficiently and accurately than in the past.

As stated by my manager, automation of this previously manual process can now save the company upwards of 40 hours per work week after taking the large number of employees who perform this task into account. Now a process that used to take 20 to 30 minutes each can now be done in a matter of seconds, and even minimizes the possibility of human error now that the data handling has all become digitized, greatly increasing the productivity of employees who dealt with discovery sessions on a regular basis.

5. CONCLUSION

The problem that ScienceLogic previously faced costed the company a substantial number of hours per work week that could have been better spent on other projects. Unfortunately, it was an issue that would have been difficult to predict when the company was in its early stages. Via automation, the tool can handle large amounts of data that caused back-end employees to struggle in terms of time spent and meticulousness. As the company continues to grow, transferring discovery sessions between SL1 systems by hand became a bottleneck regarding time efficiency. Using the tool, ScienceLogic can mitigate the time spent

replicating discovery sessions manually and avoid the possibility of human error during the process.

6. FUTURE WORK

The exporting tool was originally designed for use within the command line interface. Future versions could potentially have a user interface, which was a prospective stretch goal added to the project after its initial completion but was not included in its final version.

During development, more streamlined ways to handle the data during transfer were discovered which would allow for importing and exporting to be done without downloading a new file to the local system. By doing so, all of the data could be contained within the program during runtime to avoid taking up extra storage space. Some pieces of the program were updated to incorporate this attribute and others were kept in their more rudimentary stages, so further work could be done to optimize storage space.

The exporting tool can be used in the future to save company time, but can also be a useful tool for its application in software testing. For instance, discovery session-related errors would occur in an SL1, and a testing team would need to replicate that error on another system during error handling. With the tool, this part of resolving errors would become more efficient.

7. UVA EVALUATION

The most useful coursework applicable to the project would be the techniques taught in Algorithms (CS 4102) and Computer Architecture (CS 3330) in terms of their sections that involved optimizing loops, which were often involved during the parsing of the data contained in a discovery session.

A suggested area of improvement would be to have a more robust section on APIs, which many students first learn about in Advanced Software Development (CS 3240), but it felt

more like a brief introduction than a thorough section topic. During the internship, I learned more about working with an API and now have a better understanding of them after the project's completion.

8. ACKNOWLEDGMENTS

I would like to acknowledge my manager and supervisor, George Chikhradze, who organized an agenda as well as providing guidance during development of the tool. Additionally, I would like to recognize Caleb Garcia, a fellow intern at ScienceLogic, who assisted me in navigating the SL1 API, which is widely known within the company to be troublesome.

REFERENCES

- [1] contract.fit. 2021. 5 Companies Using Intelligent Automation to Improve Performance (May 2021). Retrieved from <https://contract.fit/academy/intelligent-automation/companies/>
- [2] Uzialko, A. 2022. Workplace Automation Is Everywhere, and It's Not Just About Robots (June 2022). Retrieved from <https://www.businessnewsdaily.com/9835-automation-techworkforce.html>
- [3] Flinders, K. 2021. HSBC standardizes on DevOps platform (Nov 2021). Retrieved from <https://www.computerweekly.com/news/252508925/HSBC-standardises-on-DevOps-platform>
- [4] LivePerson. HSBC goes Conversational with Intelligent automation. Retrieved from <https://www.liveperson.com/resources/success-stories/hsbc-goes-conversational/>
- [5] ScienceLogic. Discovering Devices. Retrieved from https://docs.sciencelogic.com/latest/Content/Web_Monitoring_Tools/Discovery_and_Credentials/discovery_managing.htm?Highlight=discovery%20session%20interface