Addition of JSON Schema Keyword “allOf” in OPA Type Checker and the Overall Importance of Open-Source Projects

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

Julia Friedman

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

Briana Morrison, Department of Computer Science

Addition of JSON Schema Keyword "allOf" in OPA Type Checker and the Overall Importance of Open-Source Projects

CS4991 Capstone Report, 2022
Julia Friedman
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
jaf9zd@virginia.edu

Abstract

While Open Policy Agent (OPA) provides an open-source engine that unifies policy enforcement across the cloud native stack, one of the limitations of its Rego type checker was the absence of the keyword "allOf." The addition of this keyword enhanced OPA's type checker, as users now receive detailed error messages regarding the use of this keyword. The "allOf" keyword, when included in an inputted JSON schema, implies that all of the fields immediately following it *must* be included in the policy being created. The addition of support for this keyword allows the type checker to recognize it and subsequently inform users if there is a mistake in their policy having to do with "allOf." This extension eliminated a limitation of OPA's type checker, and it was done via open-source code contributions. Future work could involve adding support for other keywords that are not yet implemented.

1. Introduction

OPA provides a declarative language that allows for specification of policy as code. A policy is essentially a set of rules that dictates the behavior of a software service. Policies may outline important details regarding names of trusted servers, how to comply with technical requirements, how to work within technical constraints, among many others. For example, a policy for a banking system could describe which accounts a user can withdraw money from.

Policy management across an organization is beneficial in that an administrator can manage different aspects of a system in a flexible and centralized way, thus increasing efficiency and convenience. Policy management systems allow an organization to regulate and update rules without having to alter the original implementation code because policy is managed separately from application code.

In particular, OPA is an example of a policy management service that allows for the decoupling of policy from a software service. OPA allows an organization to edit, read, write, analyze, distribute, and manage policy *separately* from the service itself (creating no need to recompile or redeploy any application code). The independence of policy helps ensure that software services can be built at scale, and makes them adaptable to changing business requirements.

2. Related Works

As explained by Corrado (2009), open-source projects are very important and beneficial for developers and organizations alike because they provide source code that is free and may be copied or altered by anyone [1]. Because OPA is an open-source project, it could be improved through the addition of support for the "allOf" keyword as previously described. In the future, OPA (and all other open-source projects) can undergo

perpetual enhancement by developers anywhere and at any time.

Von Krogh and Spaeth (2007) present the advantages of the accessibility provided by open-source projects, including its value in promoting research. Essentially, it opens the door to an abundance of data for researchers and opportunities for developer improvement of code [2]. Open-source also allows for continuous improvement of products and services, and for an open conversation about the functionality and enhancement of these software. A well-known example of an open-source product is the programming language, Python. Python is available for all users for free, allowing it to be easily distributed and broadly accessible. A user can download any source code within Python and further modify and distribute their version of that code. Overall, open-source encourages collaboration, community, and transparency among developers.

Since OPA is an open-source project, it is readily available and free to those organizations who want to use it for their needs, whether for personal or commercial application policy management. In fact, many well-recognized companies use OPA, including Google, Netflix, Atlassian, and Capital One [3].

3. Process Design

The three main components of the design of this project were comprehension of OPA's code base and its limitations, using JSON schemas for code structure validation, and understanding the relationship between OPA's Rego type checker and its error messages.

3.1. Code Base

Completion of this project required a strong understanding of OPA's code base that is publicly available on GitHub. The first steps of this project were reading through and determining what each part of the code does. The limitations of the code were already defined in the Issues tab within the OPA GitHub repository, and one of these issues was the absence of implementation for several keywords in the type checker. The goal of this project then became adding support for the "allOf" keyword.

3.2. JSON Schemas

In order to implement a new keyword, it was necessary to understand how the other keywords are defined and tested and further to understand in what location in the code itself keyword definition and testing was implemented; achieving this understanding relied on coupling the use of JSON schemas and OPA's type checker. Schemas are a helpful feature of policy management systems in order to perform type checking. A JSON (JavaScript Object Notation) schema is a tool for validating the structure of data, outlining what input should look like. Schemas are typically passed in as input when evaluating a piece of policy code, and the associated type checker will use that schema as a blueprint to authenticate the structure of the code and inputs. In the case of OPA, the JSON schemas passed in as input during evaluation can interact with code written in Rego, which is a declarative language utilized by OPA for policy writing. When these schemas are included in the evaluation input, the Rego type checker uses them as a guideline for the intended code structure.

3.3. Type Checker

With a JSON schema as a guideline, the type checker can subsequently give more detailed and helpful error messages, as it can compare the usage in the actual policy code to that defined in the schema. As implementation is added for additional keywords in the Rego type checker, the type checker gains the

ability to give specific error messages that indicate if the user has made a mistake involving *those particular* keywords. For example, if a developer made a typo involving the use of "allOf" in their inputted schema, the type checker could tell them that the root of that error relates to "allOf", instead of generally telling them that there was an error. Developers leveraging OPA can still use keywords that are not yet implemented in the type checker, but any type errors that they make in using such keywords would result in difficult-to-debug output.

3.4. Incorporation

The final steps of this project were to combine the knowledge of the code base, JSON schemas, and OPA's type checker in order to create a solution and add code for implementation of "allOf." After testing, this code was submitted as a Pull Request into the main branch of the OPA GitHub repository; this Pull Request was approved by the maintainers of OPA and was thus incorporated into the open-source, publicly available code base.

4. Results

The results of this project included a contribution into the OPA open-source project that enhanced its type checker by adding support for an additional keyword. The completion of this project helps OPA as an organization because it is an instance of the continuous improvement that is possible because OPA is open-source. The creators of OPA did not have to do any work (aside from approving the Pull Request in GitHub) in order to improve their product.

This work is also important for the many organizations that utilize OPA for their policy management needs. Companies such as Google, Netflix, Atlassian, and Capital One all use OPA for their business needs; these companies benefit because the software that they are using has now been enhanced. The specific realized improvement allows for easier and more immediate type checking regarding the use of "allOf," and thus provides more precise and informative error messages for users in these organizations who write policy using this keyword. This saves developers time and effort in debugging, and saves their employers money.

This project was done as part of an internship at IBM. Note that although IBM does not yet use OPA for policy management, the direct benefit of this work will emerge if and when IBM initiates OPA use in the future.

5. Conclusion

This technical work made meaningful contributions to the open-source community as well as the potential users of open-source products such as OPA. This work will benefit companies that utilize OPA for their policy management needs; they will have a more efficient debugging process, as implementation was added for an enhancement to the type checker that improves the quality of error messages. Improvements to OPA are made possible because of OPA's open-source nature. It is important to recognize the value in developers being able to freely contribute relevant enhancements to any open-source code base and associated product or service.

6. Future Work

Future work could involve adding support for additional keywords in the OPA type checker. For example, the "with" keyword is still unimplemented. Additionally, there are many more limitations of the code base listed under the Issues tab in the OPA GitHub repository. Because open-source code is freely available and editable, any developer can make a contribution (pending Pull Request approval by code maintainers) in

order to eliminate more existing limitations of OPA.

REFERENCES

[1] Corrado, E.M. 2009. The importance of open access, open source, and open standards (June 2009). Retrieved September 22, 2022 from http://codabox.org/15/1/istl.pdf

[2] Von Krogh, G. and Spaeth, S. 2007. The open-source software phenomenon: Characteristics that promote research. (August 2007). Retrieved September 22, 2022 from https://www.sciencedirect.com/science/article/abs/pii/S096386870700025X

[3] Open-Policy-Agent. 2022. OPA/adopters.md at main · open-policy-agent/OPA. (August 2022). Retrieved September 22, 2022 from https://github.com/open-policy-agent/opa/blob/main/ADOPTERS.md