

Automated Example Oriented REST API Documentation at Cisco

S M Sohan
Security Group
Cisco Systems Ltd.
Calgary, Canada
sosohan@cisco.com

Craig Anslow
Department of Computer Science
University of Calgary
Calgary, Canada
canslow@ucalgary.ca

Frank Maurer
Department of Computer Science
University of Calgary
Calgary, Canada
frank.maurer@ucalgary.ca

Abstract—API documentation presents both a problem and an opportunity for API usability. Representational State Transfer, or more commonly known as REST based APIs are used by software developers to interconnect applications over HTTP. Developers are required to publish and maintain the documentation of their REST APIs so that other developers can learn and use the APIs as intended. This poses the problem of identifying an efficient and effective process of generating and maintaining the documentation of REST APIs. In this paper, we have discussed our lessons learned from a case study comprising of the production use of an automated example oriented REST API documentation approach using a tool called SpyREST at Cisco over a period of eighteen months. We have observed that continuously updated documentation can be achieved by using automated test code against a REST API. Practitioners can leverage the insights shared in this paper to improve the state of their REST API documentation process. Researchers and tool developers can incorporate the ideas from this case study to extend the example oriented documentation approach to APIs beyond the realm of REST APIs.

Keywords—API; REST; Documentation; Tool; Case study; Test; Automation; HTTP; Web API;

I. INTRODUCTION

Application Programming Interfaces, commonly known as APIs, are used to express a software component in terms of its operations, inputs, outputs, and their types¹. Robillard describes API as follows: An API is the interface to implement functionality that developers can access to perform various tasks [?] [?].

REST APIs are a subclass of APIs that use standard web technologies for interconnectivity over HTTP. Fielding defined Representational State Transfer or REST as an architectural style for developing distributed hypermedia systems [?]. REST APIs provide an abstraction of the underlying system by using system specific resources such as documents, images, etc. and unique identifiers to access the resources. Using REST APIs, systems perform actions on the resources by transferring a representation of the resources between various systems. For example, the GitHub REST API² has a resource called *Repository* to denote a code repository hosted on GitHub.

Researchers identified the documentation of APIs as both the primary source of information as well as the key obstacle for API usability [?]. To this regard, researchers have identified the qualities of “good API documentation” as follows: complete, correct, includes thorough explanations and code examples, provides consistent presentation and organization [?], [?]. In our previous work, we introduced a novel technique and SpyREST, an implementation, based on an HTTP proxy server to automatically intercept example REST API calls and synthesize the data to produce REST API documentation to meet the aforementioned qualities.

The case study in this paper presents an evaluation of SpyREST in the industry. SpyREST is being used in production at Cisco for the documentation of a commercial REST API of a cloud based Cyber security product that the first author of this paper is affiliated with. It provides us with a unique opportunity to analyze the impact of the industry adoption of a tool developed in research. Production usage over an eighteen month period also allows us to understand the problem and opportunities presented by SpyREST in depth. Based on our observations, the core contribution of this case study is as follows:

- **Test driven REST API documentation.** For practitioners, we discuss a technique for producing example oriented REST API documentation as a byproduct from automated API test code.
- **Evolution of API documentation.** For practitioners, we discuss a technique for maintaining the evolution of API documentation as the API evolves.
- **Implications for future work.** We show a practical evidence of the viability of using interception to transform example API calls into API documentation. Researchers can leverage this technique for the documentation of other forms of APIs beyond REST.

The remainder of this paper is organized as follows: we present a literature review to discuss the current state of research on REST API documentation. Then, we provide a brief overview of our REST API documentation technique and the tool, SpyREST, followed by a case study of using SpyREST at Cisco. Then, we discuss our lessons learned and the limitations of this case study.

¹https://en.wikipedia.org/wiki/Application_programming_interface

²<https://developer.github.com/v3/repos/#create>

II. CONCLUSION

The conclusion goes here. this is more of the conclusion

ACKNOWLEDGMENT

The authors would like to thank... more thanks here