

Web APIs - Challenges, Design Points, and Research Opportunities

Invited Talk at the 2nd International Workshop on API Usage and Evolution (WAPI '18)

Erik Wittern

IBM T.J. Watson Research Center
Yorktown Heights, New York
witternj@us.ibm.com

ABSTRACT

Web Application Programming Interfaces (web APIs) provide programmatic, network-based access to remote data or functionalities. Applications, for example, use the Google Places API to learn about nearby establishments, use the Twitter, Instagram, or Facebook API to connect users with friends and family, or use the Stripe API to accept end-user payments. Increasingly, applications themselves consist of micro-services that expose their capabilities to one another using web APIs.

In comparison to library APIs, which are a common subject of software engineering research, web APIs present unique challenges – both for providers and consumers – that are arguably much less explored yet [3]. For one, in web APIs, providers control both the API and the runtime providing the capabilities exposed by the API. In consequence, providers may extend, change, or even remove these capabilities or the API, with possibly severe effects for consuming applications. In contrast, applications typically depend on specific versions of software libraries, which can be used even as the library evolves. Being controlled by another party and invoked via network, web API consumers also have to consider and possibly mitigate varying *quality of service* (QoS) characteristics. Primarily, the availabilities and response times of web APIs change over time, possibly impacting application performance or functionality. Furthermore, the use of library APIs is eased by mechanisms like auto-complete or IDE-integrated documentation (at least in typed languages). In contrast, web APIs commonly lack machine-understandable specifications and consume and provide data in the form of strings. To correctly use a web API, developers have to familiarize with semi-structured documentation pages, often written in HTML – there are only few IDE-based error-checking approaches for web APIs yet [2]. Whereas for many programming languages central package manager services provide unified access to available libraries (think Maven for Java, npm for JavaScript, or RubyGems for Ruby), comprehensive listings of web APIs do not exist, hardening their discovery and selection.

In this talk, we outline the characteristics of web APIs causing these challenges. We discuss relevant design points, both for

providers and consumers, how these design points have been implemented by different web API paradigms in recent years, and recent attempts to bridge these paradigms [1]. Throughout the talk, we give examples of our research to address web API-related challenges. Our goal is to inspire WAPI attendees to take on some of the many research opportunities surrounding web APIs.

CCS CONCEPTS

• **Information systems** → **Web services**; • **Software and its engineering** → *Software libraries and repositories*; *Documentation*; *Software evolution*; *Software version control*;

KEYWORDS

Web APIs, Software Engineering

ACM Reference Format:

Erik Wittern. 2018. Web APIs - Challenges, Design Points, and Research Opportunities: Invited Talk at the 2nd International Workshop on API Usage and Evolution (WAPI '18). In *WAPI'18: WAPI'18: IEEE/ACM 2nd International Workshop on API Usage and Evolution*, June 2–4, 2018, Gothenburg, Sweden. ACM, New York, NY, USA, 1 page. <https://doi.org/10.1145/3194793.3194801>

Short Biography: Erik Wittern is a Research Staff Member at the IBM T.J. Watson Research Center in New York. In recent years, Erik's research evolved around web APIs, specifically their consumption from a software engineering perspective. For IBM, he built the API Harmony catalog and currently focuses on micro-service architectures, new API paradigms like GraphQL, and serverless computing. Erik is co-author of a book on benchmarking cloud services and has written numerous research papers and patents. He holds a PhD from the Karlsruhe Institute of Technology in Germany. You can learn more about Erik's work at <http://wittern.net>, <http://www.apiful.io>, or on Twitter @erikwittern.

REFERENCES

- [1] E. Wittern, A. Cha, and J. A. Laredo. 2018. Generating GraphQL-Wrappers for REST(-like) APIs. In *Web Engineering*. Springer International Publishing. To appear.
- [2] E. Wittern, A. T. T. Ying, Y. Zheng, J. Dolby, and J. A. Laredo. 2017. Statically Checking Web API Requests in JavaScript. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*. 244–254.
- [3] E. Wittern, A. T. T. Ying, Y. Zheng, J. A. Laredo, J. Dolby, C. C. Young, and A. A. Slominski. 2017. Opportunities in Software Engineering Research for Web API Consumption. In *2017 IEEE/ACM 1st International Workshop on API Usage and Evolution (WAPI)*. 7–10.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
WAPI'18, June 2–4, 2018, Gothenburg, Sweden
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5754-8/18/06.
<https://doi.org/10.1145/3194793.3194801>