dataWASHES - Towards an Application Programming Interface of WASHES proceedings data

Allysson Allex Araújo[1], Isaac Farias[1], Victor Gonçalves[1], Rodrigo Santos[2], Davi Viana[3] e Igor Steinmacher[4]

[1]Universidade Federal do Cariri (UFCA)
Centro de Ciências e Tecnologia (CCT)
Juazeiro do Norte, Ceará – Brasil

[2]Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Rio de Janeiro, Rio de Janeiro – Brasil

[3]Universidade Federal do Maranhão (UFMA)
São Luís, Maranhão – Brasil

[4]Northern Arizona University (NAU)
Flagstaff, Arizona – Estados Unidos

allysson.araujo@ufca.edu.br,

{alves.isaac, victor.lima}@aluno.ufca.edu.br,

rps@uniriotec.br, davi.viana@ufma.br, igor.steinmacher@nau.edu

Abstract. *In recent years, there has been an increase in open science initiatives in Software Engineering research, highlighting the relevance of knowledge sharing. This paper embraces this scientific movement by introducing dataWASHES, an open source Application Programming Interface (API) that aims to facilitate streamlined programmatic access to the Workshop on Social, Human, and Economic Aspects of Software (WASHES) proceedings. By alleviating the manual data retrieval challenges and burden, dataWASHES seeks to foster collaboration and enhance research efficiency within the WASHES community. The paper delineates our API's design, implementation, and impact on WASHES knowledge exchange. Our primary contribution lies in offering a systematic tool for accessing and analyzing WASHES proceedings data, with the potential to pave the way for other research communities that eventually share a similar interest.*

1. Introduction

The increasing momentum of open science initiatives, particularly in Software Engineering (SE) research, unravels the importance of making research artifacts accessible to the public. According to Mendez *et al*. (2020), open science encompasses the disclosure of software source code (open source), data (open data), analysis scripts (open material), and study manuscripts (open access). This transparency is believed to enhance the reproducibility and replicability of scientific processes [Fernández et al. 2019].

Maedche *et al*. (2024) discussed the potential of open science in broadening global knowledge and increasing access to innovation. Hence, one can argue that open science accelerates knowledge and innovation, fostering sustainable development and resilience. As stated by UNESCO, open science encompasses all scientific disciplines and scholarly practices and is based on the following key principles: open access to scientific

knowledge, development of open science infrastructure, effective science communication, active involvement of societal actors, and open dialogue with other knowledge systems [UNESCO 2021].

In particular, Open Infrastructure refers to the technological tools and services supporting open science practices [UNESCO 2021]. This infrastructure encompasses digital platforms and repositories facilitating the dissemination, preservation, and accessibility of research outputs such as publications, data, software, and hardware. Sellanga (2023) advocates that Open Infrastructure encourages collaboration and the adoption of open standards, ensuring transparency, reusability, reproducibility, and long-term sustainability in scientific research. In addition, these infrastructures often stem from community-building efforts, emphasizing their not-for-profit nature and guaranteeing permanent, unrestricted access to the public to the greatest extent feasible [Kags 2023].

Given the importance of the Workshop on Social, Human, and Economic Aspects of Software (WASHES) in the Brazilian SE research landscape, there is a compelling opportunity to work towards an open infrastructure to streamline access to its data. With the symbolic upcoming 10^{th} edition in 2025 and a substantial archive of published papers, spanning various topics and authored by researchers from diverse backgrounds and regions across Brazil, there is immense potential for facilitating programmatic access to this valuable resource. Currently, the WASHES proceedings are openly available and well-maintained through SBC OpenLib (SOL[1]), albeit with manual access only. In this sense, this manual retrieval process can be inefficient, especially for those seeking to conduct secondary studies or robust analyses on WASHES data. To address this gap, developing an open infrastructure tailored for WASHES data would be beneficial, especially being something made *by* the community *for* the community.

Therefore, we present in this paper a preliminary version of dataWASHES[2]: a public, academic, and open source Application Programming Interface (API) designed to streamline the programmatic process of gathering data from the WASHES proceedings open available at SOL. Currently, all documents published in SOL[3] are made available under the Creative Commons license (CC BY 4.0), allowing for copying and redistribution of the material in any medium or format for any purpose. Hence, by introducing our API, in the form of open infrastructure, we aim to provide the community with a convenient tool for systematically and programmatically accessing data (papers, authors, and editions) from the proceedings, thereby enhancing openness, usefulness, and efficiency. Additionally, this paper serves as a platform for obtaining feedback from the community.

The primary contribution of this paper lies in addressing the need for a systematic, programmatic, and efficient approach to gathering data from WASHES proceedings regarding advanced search and analysis. By providing an unprecedented tool to access data in a structured and automated manner, dataWASHES could streamline the research process, fostering collaboration and knowledge exchange within the community in Brazil and beyond, thus advancing the goals of open science and open infrastructure. Furthermore, based on our future lessons with dataWASHES, we can highlight the potential to extend our proposal to other research communities that eventually share a similar interest.

[1]https://sol.sbc.org.br/index.php/washes

[2]https://gesid.github.io/dataWASHES

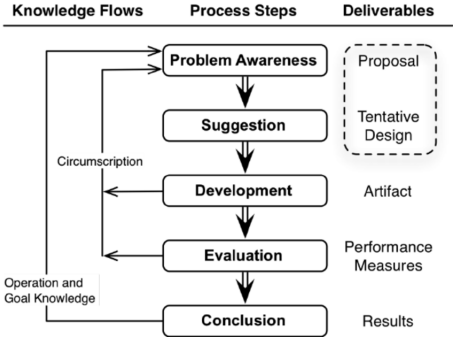
[3]https://sol.sbc.org.br/index.php/indice/faq

2. Research Design

This ongoing study is driven by descriptive research, employing a descriptive case study format [Runeson and Höst 2009]. In this context, the case study examined is WASHES. This methodological approach aligns with the paper's objective: to introduce an initial version of dataWASHES, a public and open source API to facilitate programmatic access to data from the WASHES proceedings.

Our research follows the Design Science Research (DSR) methodology [Peffers et al. 2007]. Following the protocol proposed by Kuechler and Vaishavi (2004) for DSR, a methodological plan was established, divided into five main steps (see Figure 1). We have completed the first cycle of process steps, except the Evaluation. Our next step involves conducting a second cycle, where we hope to gather feedback from the WASHES community in person and, subsequently, carry out the Evaluation (as we clarify ahead).

Figure 1. DSR methodological plan [Kuechler and Vaishnavi 2004].



The **Problem Awareness** involved analyzing both scientific and grey literature on the proposal. Specifically, we aimed to understand the nuances of Open Science and Open Infrastructure [Mendez et al. 2020, UNESCO 2021]. Regarding the API, our goal was to grasp technical aspects related to its design and implementation. Additionally, we aimed to identify current limitations in the integrated search provided by SOL and how our proposed API could mitigate this issue.

In the **Suggestion** step, we aimed to conceptualize and structure the API tentative design following the Representational State Transfer (REST) architectural style. This process involved defining the resources, endpoints, data models, and interactions. The architectural model served as a blueprint for the subsequent development phase, providing a roadmap for implementing the API.

During the **Development** step, the architectural design was transformed into a functional API (our artifact) through software coding. A key decision for this task was the selection of Flask[4], a micro web framework written in Python. Flask was chosen for its simplicity and flexibility, which significantly contributed to the ease of development. The development tasks encompassed coding the endpoints, managing requests and responses, integrating with the data source, and ensuring the API's robustness.

Evaluation will encompass qualitative and quantitative examinations. In the qualitative analysis, we will evaluate how effectively the API meets identified use cases and

[4]https://flask.palletsprojects.com

addresses technical challenges. This process will be achieved through scenarios and user interviews, providing valuable insights into the API's usefulness, openness, and potential enhancements. Computational experiments will be conducted to measure performance efficiency metrics (throughput, latency, and response time) for our quantitative analysis.

Finally, the last step is the **Conclusion**, in which all acquired knowledge is synthesized and shared alongside research findings. To achieve this goal, we plan to explore WASHES events and utilize social media to involve the community and raise awareness.

3. API Design and Implementation - A Preliminary Proposal

We designed the API to facilitate open access to the existing WASHES proceedings data. Following the Representational State Transfer (REST) architecture, our API employed the HTTP protocol for data transmission, effectively functioning as a web service. In this client-server communication paradigm, clients send HTTP requests to our API server and receive encoded data in HTTP responses.

In designing our API, we structured JSON files to efficiently organize data about papers, authors, and editions. To this end, we first manually organized a collaborative spreadsheet with all required data from WASHES proceedings. Then, we began populating the JSON files. Papers were detailed with key information such as authors (in an array format), unique paper identifier, title, publication year, abstracts in both English and Portuguese, associated keywords, paper type, and a downloadable link. Each author was represented with pertinent attributes, including name, affiliation, state, and a unique identifier, along with an array listing the IDs of papers they authored. Similarly, each edition was characterized by essential details such as its year, unique identifier, title, location, date, URL for accessing proceedings, and a list of chairs, each identified by their name, institution, and state. These JSON structures facilitated users' seamless access and retrieval of information through API endpoints, ensuring ease of interaction and utilization.

Our API was structured with endpoints representing various types of information, also referred to as "resources". The API distinguishes between three primary resources corresponding to distinct domain entities: the **Papers**, encompassing endpoints for retrieving and managing paper-related information; the **Authors**, providing endpoints for accessing author details and their associated papers; and the **Editions**, offering endpoints for retrieving edition details and lists of included papers. These endpoints enabled different types of queries and operations on the respective resources, empowering users to interact with the API and retrieve data (as a `.json` file) according to their needs. A comprehensive list of these endpoints and their respective operations were available in our supporting repository[5]. For clarity and space, we provide illustrative examples of Authors' endpoints below (the logic is quite similar for the remaining endpoints).

- Retrieve All Papers:
  - Endpoint: `GET /papers`
  - Description: Returns a list of all papers available in the API.
- Retrieve Papers by Year:
  - Endpoint: `GET /papers?year=specific_year`
  - Description: Retrieves all papers published in a specific year.

[5]https://github.com/gesid/dataWASHES

- Retrieve Papers by Author:
  - Endpoint: `GET /papers?author=specific_name`
  - Description: Retrieves all papers written by a specific author.

Finally, the implementation of our API was facilitated by Flask, which provided the necessary infrastructure for handling HTTP requests and responses as well as the development and deployment of the API. The implementation process involved coding the endpoints, handling data retrieval and manipulation, and ensuring proper error handling. Overall, our API design and implementation endeavored to provide a user-friendly and efficient means of accessing and interacting with available data, fostering transparency, collaboration, and innovation within the public data ecosystem addressed by WASHES.

4. Final Remarks and Next Steps

This paper presented the preliminary version of dataWASHES, a public, academic, and open source Application Programming Interface (API) designed to streamline the programmatic process of gathering data from the WASHES proceedings. As a contribution, dataWASHES eases data access to enhance collaboration and align with open science and open infrastructure goals. Moving forward, we plan to refine the API further, incorporating feedback from the community to improve its functionality. In this sense, we also intend to accomplish second cycle of our Design Research Science process, where we hope to gather feedback from the WASHES community in person and, subsequently, carry out the Evaluation and Conclusion. Lastly, one can highlight the potential to extend our proposal to other research communities that eventually share a similar interest about open science practices. Our initiative is open to collaboration and forming partnerships.

References

Fernández, D. M., Monperrus, M., Feldt, R., and Zimmermann, T. (2019). The open science initiative of the empirical software engineering journal. *Empirical Software Engineering*, 24:1057–1060.

Kags, A. (2023). Building community driven open infrastructure. In *Open Infrastructure Fund/Fondo de Infraestructura Abierta*.

Kuechler, B. and Vaishnavi, V. (2004). Design science research in information systems. *Available at: http://desrist.org/design-research-in-information-systems*.

Maedche, A., Elshan, E., Höhle, H., Lehrer, C., Recker, J., Sunyaev, A., Sturm, B., and Werth, O. (2024). Open science: Towards greater transparency and openness in science. *Business & Information Systems Engineering*, pages 1–16.

Mendez, D., Graziotin, D., Wagner, S., and Seibold, H. (2020). Open science in software engineering. *Contemporary empirical methods in software engineering*, pages 477–501.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77.

Runeson, P. and Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14:131–164.

Sellanga, J. (2023). Accelerating adoption and investment in open infrastructure worldwide through collaboration with networks. *Invest in Open*. Accessed: 2024-03-14.

UNESCO (2021). *UNESCO Recommendation on Open Science*. United Nations Educational, Scientific and Cultural Organization.