Do Design Rules have an Impact on the Understandability of RESTful APIs? A Controlled Experiment

Master Thesis

# Organization

|  |  |
| --- | --- |
| Examiner: | Prof. Dr. Stefan Wagner |
| Supervisors: | Dr. Justus Bogner |
| Student: |  |
| Timeframe: | 2021-05-10 – 2021-11-10 |

# Context & Motivation

RESTful web services and APIs [1] are popular in industry and represent one commonly used way to expose functionality via a well-defined and technology-agnostic interface. While there is no standardized set of design rules for REST APIs, different publications [2, 3, 4] and websites [5, 6] suggested rules and guidelines to help software developers build clean REST APIs. Another study [7] investigated which of these rules are perceived as important by industry experts. Additionally, the impact on software quality attributes was examined. Maintainability and usability were among the most associated quality attributes. Both QAs are influenced by understandability. However, this impact has not been empirically evaluated. Therefore, more research is needed to support that these design rules have in fact an impact on the understandability of RESTful APIs.

# Objectives

The goal of this study is to empirically evaluate the impact of RESTful API design rules on understandability, especially for rules considered as important by industry experts. The results should provide more insights into which rules significantly affect understandability in a positive (or even negative) way. Analyzing combinations of rules is also possible. More detailed research questions should be defined by the student.

# Methods

The research should primarily be conducted as a controlled experiment [8, 9]. Therefore, several REST API resources – some that follow different rules (treatment) and some that do not (control) – should be designed. Based on these, suitable comprehension tasks should be constructed. The tasks are then given to the participants of the experiment to work on. Suitable comprehension metrics could be correctness, time to provide a solution, subjective ratings, etc. Finally, the results of the experiment should be analyzed and documented to answer the research questions. The detailed study design should be created by the student.

# References

[1] C. Pautasso, “RESTful web services: Principles, patterns, emerging technologies,” in Web Services Foundations, vol. 9781461475, A. Bouguettaya, Q. Z. Sheng, and F. Daniel, Eds. New York, NY: Springer New York, 2014, pp. 31–51.

[2] MASSE, Mark. REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces. " O'Reilly Media, Inc.", 2011.

[3] PALMA, Francis, et al. Semantic analysis of restful apis for the detection of linguistic patterns and antipatterns. International Journal of Cooperative Information Systems, 2017, 26. Jg., Nr. 02, S. 1742001.

[4] PETRILLO, Fabio, et al. Are REST APIs for cloud computing well-designed? An exploratory study. In: International Conference on Service-Oriented Computing. Springer, Cham, 2016. S. 157-170.

[5] <https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/>

[6] <https://restfulapi.net/resource-naming/>

[7] S. Kotstein and J. Bogner, "Which RESTful API Design Rules Are Important and How Do They Improve Software Quality? A Delphi Study with Industry Experts," in Service-Oriented Computing, SummerSOC 2021, Communications in Computer and Information Science, Springer, Cham, 2021

[8] A. Jedlitschka, M. Ciolkowski, and D. Pfahl, “Reporting Experiments in Software Engineering,” in Guide to Advanced Empirical Software Engineering, London: Springer London, 2008, pp. 201–228.

[9] C. Wohlin, P. Runeson, Höst, Martin, M. C. Ohlsson, Regnell, Björn, and Wesslén, Anders, Experimentation in Software Engineering, vol. 9783642290. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 1–236.