

1 Objective

To evaluate the cost-effectiveness, quality and perceived usability of an automated REST API documentation technique using an HTTP proxy server. The goals are described as follows:

- Cost-effectiveness: The goal here is to evaluate the relative cost-effectiveness of the proposed technique vs. alternatives so that API developers can choose an appropriate technique.
- Quality: Quality of the REST API documentation is key to API learnability. API developers will be able to decide if they should use the proposed technique based on how the quality of the documentation that it generates compared against the alternatives.
- Perceived Usability: Because both the proposed technique and the generated API documentation are meant to be used by the software developers that generate or use REST API documentation. Hence, an evaluation of the usability of the proposed technique and the generated documentation will help REST API developers.

Table 1 shows a summary of the goals, their associated questions and metrics that can be used to answer the questions.

Table 1: **Goal-Question-Metric**

Goal	Question	Metrics (SpyREST vs. alternatives)
Cost-effectiveness	How much time does it take to document REST APIs?	Minutes taken to document REST APIs.
	How much input does it take to document REST APIs?	Size of input vs. output.
Quality	What is the quality of the generated REST API Documentation?	Number of examples, and HTTP headers, request and response fields included in the documentation.
Perceived Usability	How usable is the documentation tool?	Rating on a scale of 0-5 (5 being the most usable).
	How usable is the generated documentation?	Rating on a scale of 0-5 (5 being the most usable).
	What needs to be done to improve the usability?	Categorized list of suggestions from feedback.

2 Methods

The evaluation is planned as a mixed-method qualitative evaluation comprising of a case study and a user study. A prototype implementation of the proposed technique, SpyREST, will be used in the evaluation.

- **Case Study:** SpyREST is being used in a Cisco product called FireAMP for its REST API documentation. This will be used as a case-study. In particular, we'll measure the number of times the REST API documentation is auto-generated, the input to output size ratio, and a list of changes to the original SpyREST that are needed to support the specific business use-cases. We'll also report qualitative feedback about usability of the generated documentation.

In addition to this, we're also trying to find additional industry partners where SpyREST could be deployed to document their REST APIs.

- **User Study:** A user study involving software engineers from the industry will be performed to evaluate SpyREST.

Participant Selection: The user study will involve a minimum of 15 Software engineers with a minimum of 1 year experience developing or using REST APIs. Each participant will be requested to spend 1 hour for the user study. Participants can join in-person, or over the Internet using screen share and video cameras.

Method: A mix of observation and interviews. Participants will be divided into three possibly overlapping groups to carry on three different types of tasks. Group 1 will be asked to document some RESTful API actions using any method of their choice, Group 2 will be asked to document the same API actions using SpyREST, and Group 3 will be asked to compare the documentation generated by Group 1 and Group 2. We'll likely use the open-source REST API from Basho Riak¹, a key-value database, which is documented manually². Open-source REST API allows us to see how the official documentation is generated, and by using Riak the experiment is done against an API that is currently in use by software developers. Semi-structured interviews will be used following questions from Table 1.

Data Collection: We'll collect the artifacts generated by the participants and our field notes during the observation phase and semistructured interview of the studies. We'll audio record the interviews for future reference.

Analysis: We'll collect the measurements as shown in Table 1. For the qualitative data, we'll use coding to categorize the feedback about the usability of the tool and the generated API documentation.

¹<http://docs.basho.com/riak/latest/dev/references/http/>

²https://github.com/basho/basho_docs/tree/master/source/languages/en/riak/dev/references/http

3 Threats to Validity

Internal Validity: While the case study offers unique opportunities to collect data from real usage, it does not allow comparison with alternatives since only one technique is used. The user study with one hour time for each participant to document or evaluate the documentation of unfamiliar REST APIs may not sufficiently represent real-world scenario where typically a larger timespan is used. A combination of the case study and user study is used to mitigate the effects of these threats.

External Validity: The subjects of the two studies represent a small sample size. The results may not be generalizable and statistically significant beyond the study participants. Access to study participants is a practical limitation here. The qualitative studies are designed to go in depth about the evaluation to satisfy the goals. Future research needs to be carried out to extend the studies involving a larger population of projects and REST API developers.