

# Reducing Workload in Using AI-based API REST Test Generation

Benjamin Leu  
benjamin.leu@students.fhnw.ch  
University of Applied Sciences and  
Arts Northwestern Switzerland  
Windisch, Switzerland

Jonas Volken  
jonas.volken@students.fhnw.ch  
University of Applied Sciences and  
Arts Northwestern Switzerland  
Windisch, Switzerland

Martin Kropp  
martin.kropp@fhnw.ch  
University of Applied Sciences and  
Arts Northwestern Switzerland  
Windisch, Switzerland

Nejdet Dogru  
nejdet.dogru@testifi.io  
Testifi GmbH  
Munich, Germany

Craig Anslow  
craig.anslow@ecs.vuw.ac.nz  
Victoria University of Wellington  
Wellington, New Zealand

Robert Biddle  
robert.biddle@carleton.ca  
Carleton University  
Ottawa, Canada

## ABSTRACT

Modern software applications, notably those utilizing microservices architectures, rely heavily on REST API technology for communication. Testing these APIs is challenging, time-consuming, and prone to errors. This paper introduces Pulse-UI, an AI-supported tool designed to enhance test sequence generation for REST APIs, aiming to reduce the workload involved in managing test sequences efficiently and improve overall test quality.

## KEYWORDS

Software Testing, AI-based Testing, REST API, Automation, Teaching AI System, Test Case Scenarios, Human support

### ACM Reference Format:

Benjamin Leu, Jonas Volken, Martin Kropp, Nejdet Dogru, Craig Anslow, and Robert Biddle. 2024. Reducing Workload in Using AI-based API REST Test Generation. In *Proceedings of 5th International Conference on Automation of Software Test (AST 2024)*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUCTION

Modern software development increasingly adopts microservices architectures, relying heavily on REST API technology for communication. This shift necessitates robust testing strategies to ensure API correctness, reliability and performance. The increasing adoption of REST API based software development has lead to a lot of research and development activities for automated test generation. However, existing REST API testing approaches are time-consuming, error-prone, and often result in inadequate test coverage due to the manual effort required for test management ([2, 1, 4]. AI based tools often generate incomplete or even false tests, and require a lot of effort to define and manage test scenarios. This paper introduces Pulse-UI, an AI-supported tool designed to streamline REST API test scenario generation and management, aiming to enhance test quality, and reduce manual testing workload.

---

Permission to make digital or hard copies of all or part 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 components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

AST 2024, April 2024, Lisbon, Portugal

© 2024 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM... \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 2 PULSE-UI

Pulse-UI [3] was developed to address the challenges of REST API testing by providing an intuitive scenario graph visualization for efficient test case scenario manipulation as shown in Figure 1. Leveraging AI based tools, Pulse-UI assists in generating, analyzing, and manipulating test scenarios which significantly reduces the manual effort involved in REST API testing. The Pulse-UI tool features a usable interface for visualizing API endpoints, creating and linking test sequences, and automating test data generation. Pulse-UI's core functionalities are shown in Figure 1 and include:

- **Scenario Graph Visualization:** Offers an overview and detailed view of test scenarios, facilitating easy manipulation of APIs.
- **AI-Supported Test Generation:** Utilizes AI to suggest potential test sequences and data, enhancing test coverage and efficiency.
- **Efficient Test Data Management:** Simplifies the creation, editing, and linkage of test data across different test cases, encouraging more comprehensive testing as visualized in Figures 2 and 3.

## 3 RESULTS

A preliminary user study with one developer demonstrated Pulse-UI's effectiveness in reducing the time and effort required to generate and maintain REST API test sequences. Compared to manual testing methods and existing tools like Postman and Swagger UI, *Pulse-UI* significantly reduced test creation time, especially for complex scenarios. The study highlighted the following key outcomes:

- **Time Efficiency:** It was reported that Pulse-UI saved considerable time in creating both simple and complex test sequences. The tool's suggestion system and intuitive interface streamlined the test generation process.
- **Enhanced Test Quality:** By facilitating the easy creation of linked test sequences and automating test data generation, Pulse-UI enabled more thorough testing of edge cases and complex scenarios.
- **User Experience:** Feedback from the study praised Pulse-UI for its effective user interface and efficient workflow, noting its superiority in usability and efficiency compared to manual testing methods.

*Limitations.* A major limitation of the work is the very limited validation with only one test participant. Our design may unknowingly have been biased to this particular participant, their background, and their views on testing. Another issue might be that the main focus of this project was on developing a user interface to improve efficiency in manual editing and management of REST test case scenarios. We did not yet evaluate the effectiveness of the application with respect to test quality improvement and efficiency of learning of the AI engine.

Name	Description	Sequence	Actions
Pet Testsequence	Test API Sequence creation Speed of the Pulse Sequence Creator compared to other API testing methods	<ul style="list-style-type: none"> <li>PUT /pet</li> <li>POST /pet</li> <li>GET /pet/{petId}</li> </ul>	[Edit] [Share] [Delete]
Store Testsequence	Sequence to test the store endpoints	<ul style="list-style-type: none"> <li>GET /store/inventory</li> <li>POST /store/order</li> <li>DELETE /store/order/{orderId}</li> </ul>	[Edit] [Share] [Delete]
User Testsequence	Test user endpoints of pet store	<ul style="list-style-type: none"> <li>GET /user/login</li> <li>GET /user/logout</li> <li>GET /user/{username}</li> </ul>	[Edit] [Share] [Delete]

Figure 1: The overview screen of Pulse-UI.

Sequence Editor - Pet Testsequence

Test Data:

- PetId 1: string, value: 1
- New Pet Id: string, value: 999111
- New Pet Name: string, value: Doge
- Category Name: string, value: Puppy
- Category Id: number, value: 111
- Status: string, value: Available
- New Pet Name: string, value: New Doge

Sequence:

- GET /pet/{petId} Returns a single pet (getPetById)
- POST /pet Add a new pet to the store (addPet)
- GET /pet/{petId} Returns a single pet (getPetById)
- PUT /pet Update an existing pet by Id (updatePet)

Endpoint Details (POST /pet):

Request:

```

body: {
  id: integer [New Pet ID X],
  name: string [New Pet Name X],
  category: { ... },
  photoUrls: [
    string [0]
  ],
  tags: [
    > { ... }
  ],
  status: string [Status X]: pet status in the store
}
    
```

Response:

```

body: {
  id: integer ,
  name: string ,
  category: { ... },
  photoUrls: [
    string
  ],
  tags: [
    > { ... }
  ],
  status: string : pet status in the store
}
    
```

Figure 2: The editor screen of Pulse-UI with a selected endpoint and its details.

GET /repositories/{workspace}/{repo\_slug}/branching-model/settings

Return the branching model configuration for a repository. The returned object:

- Always has a development property for the development branch.

Request

Parameters

repo\_slug: string [This can either be the repository slug or the UUID of the repository, surrounded by ...]

workspace: string [This can either be the workspace ID (slug) or the workspace UUID surrounded by ...]

Figure 3: Example of collapsed endpoint details.

## 4 SUMMARY

Pulse-UI represents a significant advancement in REST API testing, offering a solution that not only reduces the manual workload but also improves test quality. Future work will focus on expanding the tool’s AI capabilities, integrating with additional testing frameworks, further enhancing the user experience to support a broader range of testing scenarios, and conducting further user studies to understand the effectiveness of the tool.

## REFERENCES

- [1] Ovidiu Baniş et al. “Automated Specification-Based Testing of REST APIs”. In: *Sensors* 21.16 (2021). ISSN: 1424-8220. DOI: 10.3390/s21165375.
- [2] Myeongsoo Kim et al. “Automated Test Generation for REST APIs: No Time to Rest Yet”. In: *Proceedings of the 31st International Symposium on Software Testing and Analysis. ISSTA 2022*. Virtual, South Korea, 2022, pp. 289–301. DOI: 10.1145/3533767.3534401.
- [3] *Testifi Pulse*. 2023. URL: <https://testifi.io/pulse-api-testing/> (visited on 11/16/2023).
- [4] Emanuele Viglianisi, Michael Dallago, and Mariano Ceccato. “RESTTESTGEN: Automated Black-Box Testing of RESTful APIs”. In: *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*. 2020, pp. 142–152. DOI: 10.1109/ICST46399.2020.00024.