

REST in Pieces: **RESTful Design Rule Violations in** **Student-built Web Apps**

Sergio Di Meglio¹, Valeria Pontillo², Luigi Libero Lucio Starace¹

¹ *Università degli Studi di Napoli Federico II, Naples, Italy*

² *Gran Sasso Science Institute (GSSI), L'Aquila, Italy*

SEAA 2025

51st Euromicro Conference on Software Engineering and Advanced Applications
September 10-12, Salerno, Italy

REST APIs: A QUICK PRIMER

- REST (Representational State Transfer) is the **dominant architectural style** for Web APIs.

REST APIs: A QUICK PRIMER

- REST (Representational State Transfer) is the **dominant architectural style** for Web APIs.
- REST APIs rely on a set of **principles and guidelines**:
 - **Stateless communication**
 - **Standard use of HTTP methods** (GET, POST, PUT, DELETE)
 - **Consistent URI design**
 - **Clear resource modelling**

REST APIs: A QUICK PRIMER

- REST (Representational State Transfer) is the **dominant architectural style** for Web APIs.
- REST APIs rely on a set of **principles and guidelines**:
 - **Stateless communication**
 - **Standard use of HTTP methods** (GET, POST, PUT, DELETE)
 - **Consistent URI design**
 - **Clear resource modelling**
- Poor REST design:
 - Leads to **technical debt** and **increased maintenance costs**
 - **Affects usability, scalability, and developer experience.**

MOTIVATION

- Many studies investigated the quality of student-developed software, focusing on code smells and architectural antipatterns in programming assignments or object-oriented code (e.g.: [1])

[1] De Luca, Marco, et al. "Rookie Mistakes: Measuring Software Quality in Student Projects to Guide Educational Enhancement." SEAA 2025.

MOTIVATION

- Many studies investigated the quality of student-developed software, focusing on code smells and architectural antipatterns in programming assignments or object-oriented code (e.g.: [1])
- Fewer attention has been directed to quality in REST API design

[1] De Luca, Marco, et al. "Rookie Mistakes: Measuring Software Quality in Student Projects to Guide Educational Enhancement." SEAA 2025.

MOTIVATION

- Many studies investigated the quality of student-developed software, focusing on code smells and architectural antipatterns in programming assignments or object-oriented code (e.g.: [1])
- Fewer attention has been directed to quality in REST API design
- By analyzing student-built APIs, we can:
 - Identify **recurring design issues**
 - Improve **educational practices**
 - Help **industry better prepare for onboarding graduates**

[1] De Luca, Marco, et al. "Rookie Mistakes: Measuring Software Quality in Student Projects to Guide Educational Enhancement." SEAA 2025.

RESEARCH GOAL AND CONTEXT

RQ: What are the most frequent REST API design rule violations in student-developed web applications?

RESEARCH GOAL AND CONTEXT

RQ: What are the most frequent REST API design rule violations in student-developed web applications?

- **Course:** Web Technologies
 - B.Sc. in Computer Science, 3rd year, **6 ETCS** course
 - Goal is to teach modern full-stack web development

RESEARCH GOAL AND CONTEXT

RQ: What are the most frequent REST API design rule violations in student-developed web applications?

- **Course:** Web Technologies
 - B.Sc. in Computer Science, 3rd year, **6 ETCS** course
 - Goal is to teach modern full-stack web development
- **Projects:** 40 full-stack web apps (Spring 2024)

RESEARCH GOAL AND CONTEXT

RQ: What are the most frequent REST API design rule violations in student-developed web applications?

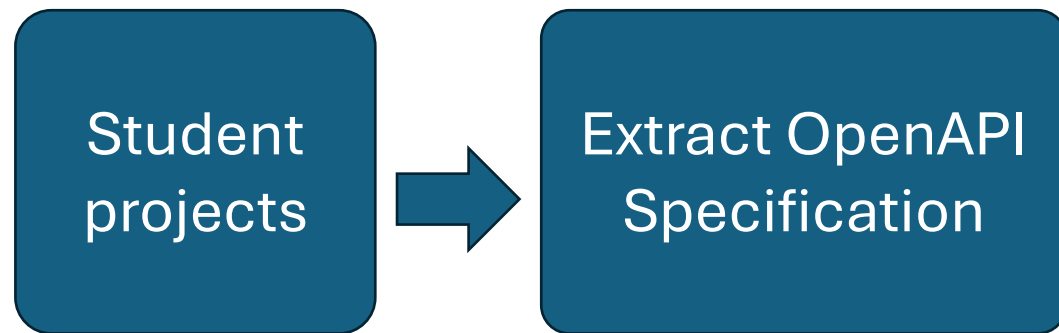
- **Course:** Web Technologies
 - B.Sc. in Computer Science, 3rd year, **6 ETCS** course
 - Goal is to teach modern full-stack web development
- **Projects:** 40 full-stack web apps (Spring 2024)
- **Tech Stack:**
 - Backend: 97.5% Express (JS), 2.5% Spring (Java)
 - Frontend: Angular (82.5%), React (15%), Vue (2.5%)

ANALYSIS PIPELINE

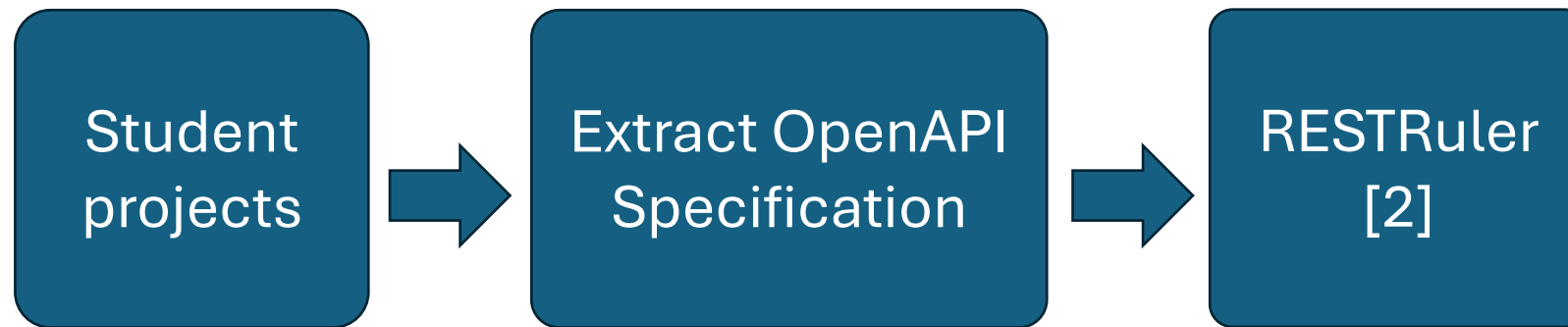


Student
projects

ANALYSIS PIPELINE

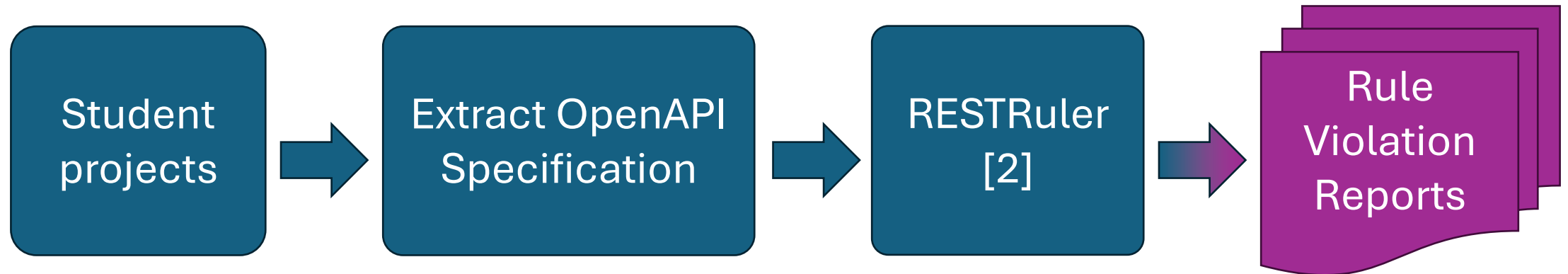


ANALYSIS PIPELINE



[2] Bogner, Justus, et al. "RESTRuler: towards automatically identifying violations of RESTful design rules in web APIs." *2024 IEEE 21st International Conference on Software Architecture (ICSA)*. IEEE, 2024.

ANALYSIS PIPELINE



[2] Bogner, Justus, et al. "RESTRuler: towards automatically identifying violations of RESTful design rules in web APIs." *2024 IEEE 21st International Conference on Software Architecture (ICSA)*. IEEE, 2024.

RESTRuler: Supported Rules

- RESTRuler [2] is capable to assess **14** different REST design rules

[2] Bogner, Justus, et al. "RESTRuler: towards automatically identifying violations of RESTful design rules in web APIs." *2024 IEEE 21st International Conference on Software Architecture (ICSA)*. IEEE, 2024.

RESTRuler: Supported Rules

- RESTRuler [2] is capable to assess **14** different REST design rules
- Examples include:

Rule description	Category
Use plural nouns for collections (e.g.: "/users")	URI Design
Avoid CRUD terms (e.g., "/getUser") in URIs.	URI Design
Avoid underscores in URIs	URI Design
Use HTTP GET method only to retrieve resources.	HTTP Semantics
Do not use GET or POST to tunnel other HTTP methods.	HTTP Semantics
Include appropriate Content-Type headers.	Metadata/Headers

[2] Bogner, Justus, et al. "RESTRuler: towards automatically identifying violations of RESTful design rules in web APIs." *2024 IEEE 21st International Conference on Software Architecture (ICSA)*. IEEE, 2024.

RESULTS: REST IN PIECES

RESULTS: REST IN PIECES

URI Design & Structure

- Violations in **URI formatting and naming** (e.g., hyphens, plural nouns, CRUD terms)
- **100% of projects** violated at least one rule in this category

RESULTS: REST IN PIECES

URI Design & Structure

- Violations in **URI formatting and naming** (e.g., hyphens, plural nouns, CRUD terms)
- **100% of projects** violated at least one rule in this category

HTTP Method Semantics

- Misuse of **GET for non-retrieval**, tunneling other methods
- **83% of projects** affected

RESULTS: REST IN PIECES

URI Design & Structure

- Violations in **URI formatting and naming** (e.g., hyphens, plural nouns, CRUD terms)
- **100% of projects** violated at least one rule in this category

HTTP Method Semantics

- Misuse of **GET for non-retrieval**, tunneling other methods
- **83% of projects** affected

Metadata & Headers

- Missing or incorrect **Content-Type** headers
- **35% of projects** had violations

INSIGHTS

- **Design quality is not intuitive:** Even with dedicated lectures, students tend to overlook design guidelines

INSIGHTS

- **Design quality is not intuitive:** Even with dedicated lectures, students tend to overlook design guidelines
- **For Educators:** Automated tools/linters like RESTRuler can uncover issues that manual review may miss, supporting scalable quality assessment.

INSIGHTS

- **Design quality is not intuitive:** Even with dedicated lectures, students tend to overlook design guidelines
- **For Educators:** Automated tools/linters like RESTRuler can uncover issues that manual review may miss, supporting scalable quality assessment.
- **Curriculum implications:**
 - Consider integrating **continuous feedback mechanisms** (e.g., linters, checklists) during development.
 - Dedicate more space to the importance of adhering to guidelines and conventions

FUTURE WORKS

- **Expand quality analysis:**
 - Integrate tools like **SonarQube** to detect code smells and technical debt.

FUTURE WORKS

- **Expand quality analysis:**
 - Integrate tools like **SonarQube** to detect code smells and technical debt.
- **Evaluate pedagogical interventions:**
 - Compare quality **before and after** introducing targeted REST design lectures.
 - Test the impact of **continuous static analysis** during development.

FUTURE WORKS

- **Expand quality analysis:**
 - Integrate tools like **SonarQube** to detect code smells and technical debt.
- **Evaluate pedagogical interventions:**
 - Compare quality **before and after** introducing targeted REST design lectures.
 - Test the impact of **continuous static analysis** during development.
- **Develop support materials:**
 - Provide students with **REST design templates, checklists, and linting tools**.

FUTURE WORKS

- **Expand quality analysis:**
 - Integrate tools like **SonarQube** to detect code smells and technical debt.
- **Evaluate pedagogical interventions:**
 - Compare quality **before and after** introducing targeted REST design lectures.
 - Test the impact of **continuous static analysis** during development.
- **Develop support materials:**
 - Provide students with **REST design templates, checklists, and linting tools**.
- **Broaden scope:**
 - Replicate the study across **multiple cohorts and institutions** to assess generalizability.

REST IN PIECES: RESTFUL DESIGN RULE VIOLATIONS IN STUDENT-BUILT WEB APPS



Luigi Libero Lucio Starace

luigiliberolucio.starace@unina.it

<https://luistar.github.io>

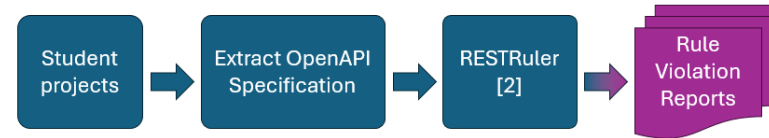
RESEARCH GOAL AND CONTEXT

RQ: What are the most frequent REST API design rule violations in student-developed web applications?

- **Course:** Web Technologies
 - B.Sc. in Computer Science, 3rd year, **6 ETCS** course
 - Goal is to teach modern full-stack web development
- **Projects:** 40 full-stack web apps (Spring 2024)
- **Tech Stack:**
 - Backend: 97.5% Express (JS), 2.5% Spring (Java)
 - Frontend: Angular (82.5%), React (15%), Vue (2.5%)

SEAA 2025 - REST in Pieces: REST Design Rule Violations in Student-built Web Apps - Luigi L. L. Starace

ANALYSIS PIPELINE



[2] Bogner, Justus, et al. "RESTRuler: towards automatically identifying violations of RESTful design rules in web APIs." *2024 IEEE 21st International Conference on Software Architecture (ICSA)*. IEEE, 2024.

SEAA 2025 - REST in Pieces: REST Design Rule Violations in Student-built Web Apps - Luigi L. L. Starace

RESULTS: REST IN PIECES

URI Design & Structure

- Violations in **URI formatting and naming** (e.g., hyphens, plural nouns, CRUD terms)
- **100% of projects** violated at least one rule in this category

HTTP Method Semantics

- Misuse of **GET** for non-retrieval, tunneling other methods
- **83% of projects** affected

Metadata & Headers

- Missing or incorrect **Content-Type** headers
- **35% of projects** had violations

SEAA 2025 - REST in Pieces: REST Design Rule Violations in Student-built Web Apps - Luigi L. L. Starace

INSIGHTS

- **Design quality is not intuitive:** Even with dedicated lectures, students tend to overlook design guidelines
- **For Educators:** Automated tools/linters like RESTRuler can uncover issues that manual review may miss, supporting scalable quality assessment.
- **Curriculum implications:**
 - Consider integrating **continuous feedback mechanisms** (e.g., linters, checklists) during development.
 - Dedicate more space to the importance of adhering to guidelines and conventions

SEAA 2025 - REST in Pieces: REST Design Rule Violations in Student-built Web Apps - Luigi L. L. Starace