

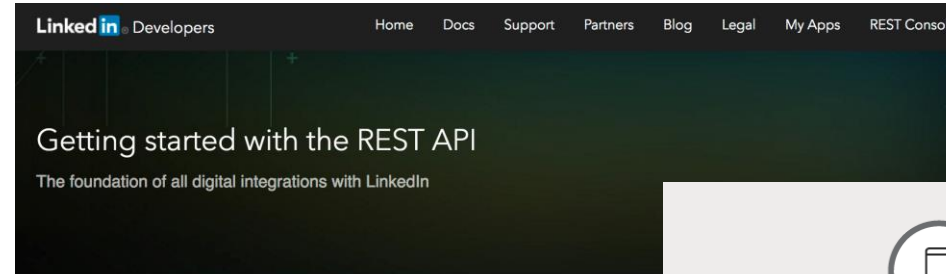
# Fuzzing Web APIs: Overview of Existing Tools

Prof. Andrea Arcuri  
Kristiania University College and  
OsloMet

# Web Services

- Providing APIs (Application Programming Interfaces) over network, remote servers
- Communications over UDP/TCP, with protocols like HTTP
- Different types of data transfer formats
  - JSON, XML, HTML, plain text, etc.
- Permanent storage:
  - eg, SQL/NoSQL databases
- **REST API**s most common type of web services
  - others are *SOAP*, *GraphQL* and *gRPC*

# REST APIs are used everywhere...



This is automatically-generated documentation for the reddit API.

The reddit API and code are [open source](#). Found a mistake or interested in helping us improve? Have a gander at [api.py](#) and send us a pull request.

Please take care to respect our [API access rules](#).

### overview

#### listings

Many endpoints on reddit use the same protocol for controlling pagination and filtering. These endpoints are called Listings and share five common parameters: `after`, `before`, `limit`, `count`, and `show`.

Listings do not use page numbers because their content changes so frequently. Instead, they allow you to view slices of the underlying data. Listing JSON responses contain `after` and `before` fields which are equivalent to the "next" and "prev" buttons on the site and in combination with `count` can be used to page through the listing.

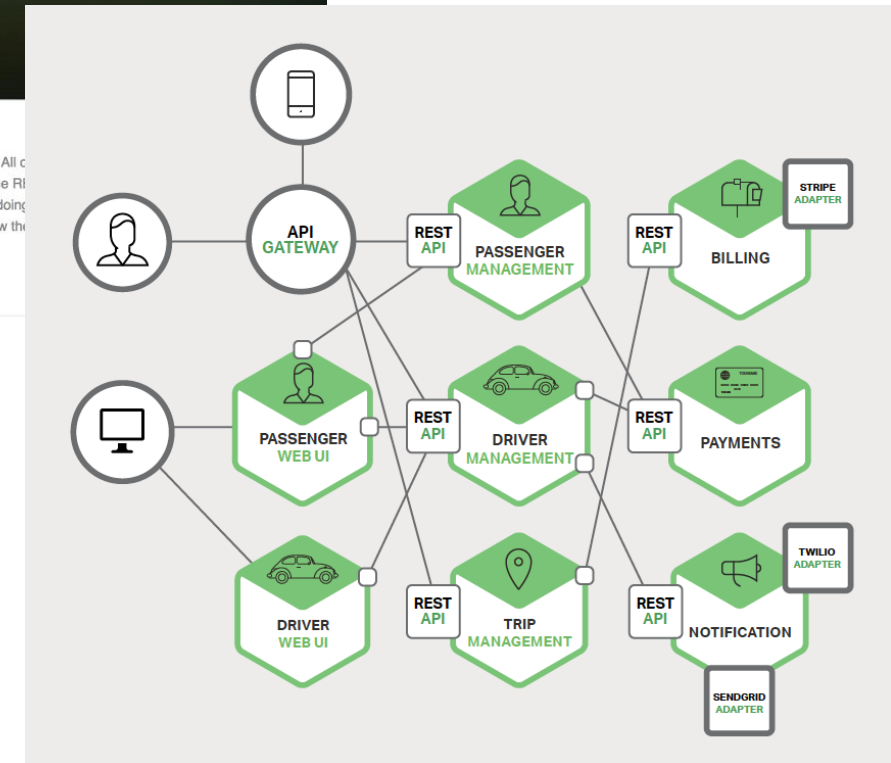
The common parameters are as follows:

- `after` / `before` - only one should be specified. these indicate the `fullname` of an item in the listing to use as the anchor point of the slice.
- `limit` - the maximum number of items to return in this slice of the listing.
- `count` - the number of items already seen in this listing. on the html site, the builder uses this to determine when to give values for `before` and `after` in the response.

Method	HTTP request	Description
get	GET /files/ <i>fileId</i>	
insert	POST https://www.googleapis.com/upload/drive/v2/files and POST /files	
patch	PATCH /files/ <i>fileId</i>	
update	PUT https://www.googleapis.com/upload/drive/v2/files/ <i>fileId</i> and PUT /files/ <i>fileId</i>	
copy	POST /files/ <i>fileId</i> /copy	

### API methods

Method	HTTP request	Description
account		
/api/v1/me	oauth	
/api/v1/me/blocked	oauth	
/api/v1/me/friends	oauth	
/api/v1/me/karma	oauth	
/api/v1/me/prefs	oauth	
/api/v1/me/trophies	oauth	
/prefs/blocked	oauth	
/prefs/friends	oauth	
/prefs/messaging	oauth	
/prefs/trusted	oauth	
/prefs/where	oauth	
captcha		
/api/needs_captcha	oauth	
flair		
/api/clearflairtemplates	oauth	
/api/deleteflair	oauth	
/api/deleteflairtemplate	oauth	
/api/flair	oauth	
/api/flairconfig	oauth	



# REST Testing Challenges

- How to choose **query** and **path** parameters?
- How to prepare **body payloads** (e.g. JSON)?
- How to choose data to insert into **SQL** databases?
- Goals:
  - **Finding faults** (eg crashes)
  - **Maximize code coverage** (eg, regression tests)
- Writing high coverage tests *by hand* for every single endpoint is time consuming

# What about **Automated Test Generation** for RESTful APIs?

- Automatically write all the test cases
- Not just execution, but choice of all the inputs
- Hard, complex problem

## 2 Uses of Generated Tests

- If automated oracles: **automatically detect faults**
  - e.g., HTTP response giving 500
- No oracles / faults: **regressing testing**
  - Tests can be added to Git, to capture current behavior of system
  - If in future introduce new bug that breaks functionality, regression tests will start to fail

# Fuzzers

- Tools that automatically generate test inputs
- Different strategies: from **random** inputs to advanced **AI** techniques
- Can automatically create and evaluate **millions** of test cases
- Used in many different domains
  - eg, parser libraries and unit testing
- REST fuzzing is a more recent development
  - eg, Restler, Schemathesis, RESTest, Fuzz-Lightyear and EvoMaster

23 tags

289 forks

Report repository



 **schemathesis** Public

💖 Sponsor

👁 Watch 20

🔗 Fork 153

☆ Star 2.2k


🔗 master

🔗 22 Branches

🏷 279 Tags

Add file

<> Code

 <b>Stranger6667</b> chore: Release 3.34.1	33025ea · 1 hour ago	🕒 2,416 Commits
📁 .github	build(deps): Bump github/codeql-action from 3.26.0 to 3.26.2	yesterday
📁 benches	feat: Add a way to replace impl	last month
📁 corpus	chore: Rework corpus	3 months ago
📁 docs	chore: Release 3.34.1	1 hour ago
📁 example	feat: Flexible operations filter	last month
📁 img	docs: Update documentation	9 months ago
📁 src/schemathesis	fix: Error in response_header_conformance if the header definit...	12 hours ago
📁 test-corpus	chore: Sort imports	2 months ago
📁 test	fix: Error in response_header_conformance if the header definit...	12 hours ago
📄 .dockerignore	docs: Update README	last year
📄 .gitignore	chore: Remove duplicate pattern from .gitignore	4 years ago
📄 .gitmodules	test: Add tests on schemas from Open API catalog	4 years ago

### About

Supercharge your API testing, catch bugs, and ensure compliance

[🔗 schemathesis.readthedocs.io](#)

testing

graphql

cli

swagger

openapi

pytest

property-based-testing

hypothesis

hacktoberfest

openapi3

📖 Readme

📜 MIT license

📜 Code of conduct

🔗 Cite this repository

📶 Activity

📁 Custom properties

☆ 2.2k stars

👁 20 watching

🔗 153 forks

Report repository

### Releases

 277

 **RESTest** Public

👁 Watch 16 ▾

🔗 Fork 34 ▾




★ Starred 205 ▾

🔑 master ▾ 🔗 45 Branches 🏷 12 Tags



🔍 Go to file 


Add file ▾

 Code ▾

 **josgarmar31** Merge pull request [#273](#) from isa-group/feature/[#272](#)   be6aab3 · 2 months ago 🕒 1,280 Commits

📁 .circleci	CIRCLECI	last year
📁 allure	Fix permissions of Allure scripts	2 years ago
📁 docs	Update README	10 months ago
📁 src	ReadProperties Modification	2 months ago
📄 .gitignore	Fix errors and refactor	10 months ago
📄 LICENSE	LICENSE updated	4 years ago
📄 README.md	Update README.md	5 months ago
📄 pom.xml	[maven-release-plugin] prepare for next development iterati...	3 months ago

📖 **README** 🛡️ LGPL-3.0 license  



### About


RESTest: Automated Black-Box Testing of RESTful Web APIs

- java testing rest rest-api swagger
- openapi api-rest oas api-testing

- 📖 Readme
- 🛡️ LGPL-3.0 license
- 📈 Activity
- 📖 Custom properties
- ★ 205 stars
- 👁 16 watching
- 🔗 34 forks

Report repository

### Releases 7

 **restdest-1.5.0** Latest

on May 20

+ 6 releases

### Contributors 9



fuzz-lightyear

Public

👁 Watch 8 ▾

🔗 Fork 25 ▾

☆ Star 205 ▾

🔗 master ▾

🔗 15 Branches

🔗 11 Tags

🔍 Go to file

t

Add file ▾

<> Code ▾

👤 Chandra158	Merge pull request #94 from Yelp/update-changelog	⋮	✅	aeaae96 · 3 months ago	🕒 226 Commits
📁 .github/workflows	Update Github action PyPi publish version				6 months ago
📁 docs	adding logo				3 years ago
📁 fuzz_lightyear	update version				3 months ago
📁 scripts	version bump; adding uploader script				5 years ago
📁 test_data	adding --ignore-non-vulnerable flag (#53)				4 years ago
📁 testing	Update generator to consider 201 as success response code				last year
📁 tests	Update generator to consider 201 as success response code				last year
📄 .gitignore	adding database support, and apikey auth to testing app				5 years ago
📄 .pre-commit-config.yaml	Add Python 3.8 support and drop 3.7				last year
📄 CHANGELOG.md	update changelog: 0.0.11				3 months ago
📄 CONTRIBUTING.md	adding documentation				5 years ago
📄 LICENSE	fix LICENSE				5 years ago
📄 Makefile	nicer cURL output				5 years ago

## About

A pytest-inspired, DAST framework, capable of identifying vulnerabilities in a distributed, micro-service ecosystem through chaos engineering testing and stateful, Swagger fuzzing.

- 📖 Readme
- 📄 View license
- 📈 Activity
- 📁 Custom properties
- ☆ 205 stars
- 👁 8 watching
- 🔗 25 forks

Report repository

## Releases

🔗 11 tags

## Packages








No packages published

Contributors 9

<> Code ▾



A word cloud of software testing and development related terms. The words are arranged in a grid-like fashion, with some words appearing in larger, more prominent bubbles. The terms include: kotlin, java, testing, graphql, rest, grpc, thrift, evolutionary-algorithms, fuzzing, api-rest, fuzzer, api-testing, rpc-api, test-case-generation, and search-based-software-testing.

-  [Readme](#)
  -  [GPL-3.0 license](#)
  -  [Activity](#)
  -  [Custom properties](#)
  -  **469 stars**
  -  **24 watching**
  -  **77 forks**
- [Report repository](#)

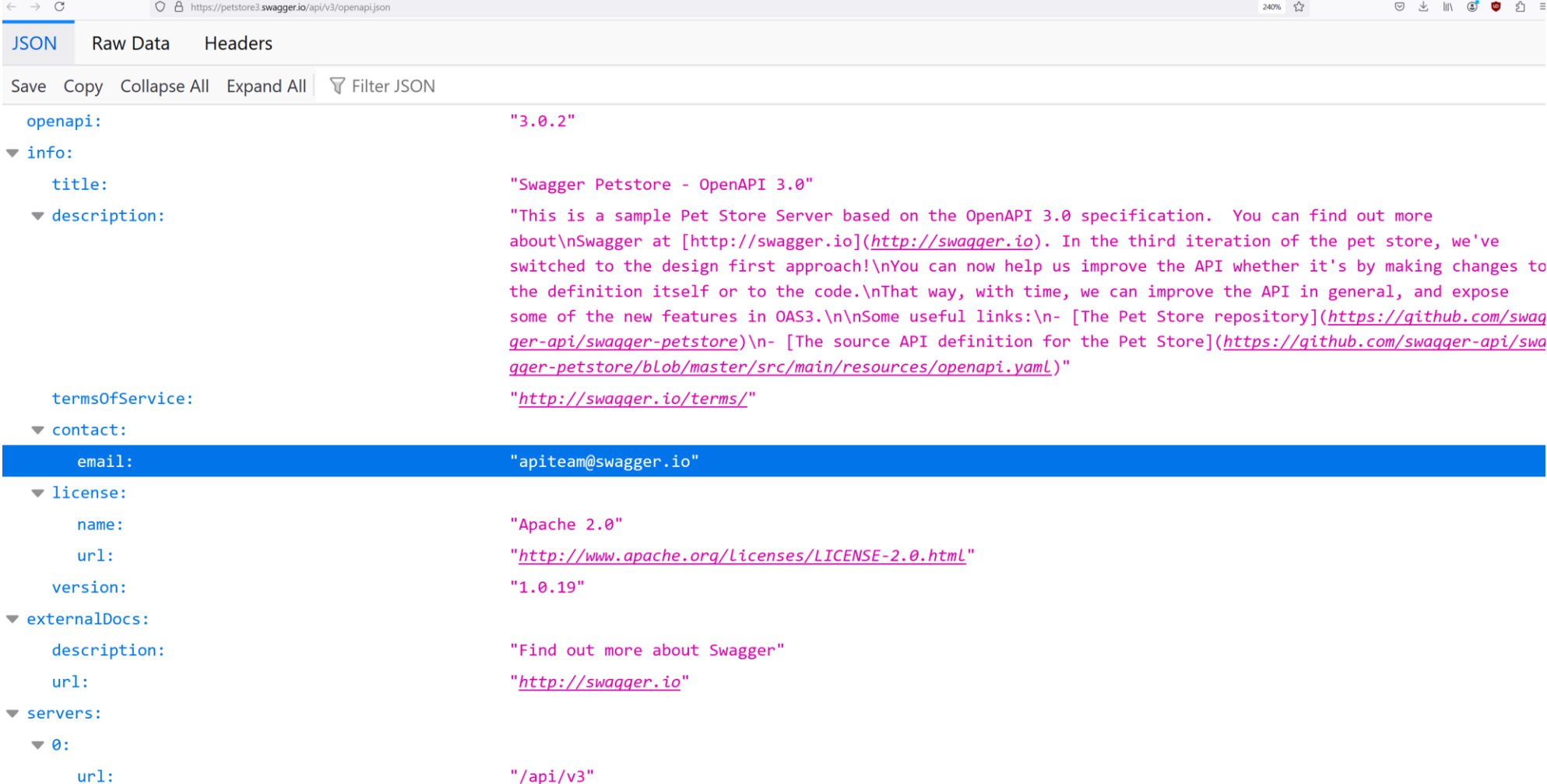
## Releases 10

# Input: OpenAPI/Swagger Schema

- Need to know what endpoints are available, and their parameters
- Schema defining the APIs
- OpenAPI is the most popular one
- Defined as JSON file, or YAML

# Example: PetStore

- Online schema at <https://petstore3.swagger.io/api/v3/openapi.json>



The screenshot shows a web browser displaying the JSON schema for the PetStore API. The browser's address bar shows the URL `https://petstore3.swagger.io/api/v3/openapi.json`. The page is rendered in a JSON viewer interface with tabs for 'JSON', 'Raw Data', and 'Headers'. The 'JSON' tab is active, and the schema is displayed in a collapsible tree view. The schema is a JSON object with the following structure:

```
{
  "openapi": "3.0.2",
  "info": {
    "title": "Swagger Petstore - OpenAPI 3.0",
    "description": "This is a sample Pet Store Server based on the OpenAPI 3.0 specification. You can find out more about Swagger at [http://swagger.io](http://swagger.io). In the third iteration of the pet store, we've switched to the design first approach! You can now help us improve the API whether it's by making changes to the definition itself or to the code. That way, with time, we can improve the API in general, and expose some of the new features in OAS3. Some useful links: - [The Pet Store repository](https://github.com/swagger-api/swagger-petstore) - [The source API definition for the Pet Store](https://github.com/swagger-api/swagger-petstore/blob/master/src/main/resources/openapi.yaml)",
    "termsOfService": "http://swagger.io/terms/",
    "contact": {
      "email": "apiteam@swagger.io"
    },
    "license": {
      "name": "Apache 2.0",
      "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
    }
  },
  "externalDocs": {
    "description": "Find out more about Swagger",
    "url": "http://swagger.io"
  },
  "servers": [
    {
      "url": "/api/v3"
    }
  ]
}
```

The 'info' object is expanded, showing the 'description' field with a detailed text block. The 'contact' object is also expanded, showing the 'email' field. The 'license' object is expanded, showing the 'name' and 'url' fields. The 'externalDocs' object is expanded, showing the 'description' and 'url' fields. The 'servers' array is expanded, showing a single server object with the 'url' field.

# What Can Expect?

- All these tools will analyze the schema
- Send requests with many different strategies
  - there is lot of research in academia on this
- Check if any error in the API can be identified
- Output executable test cases
  - in different formats, eg Java and Python

```
$ evomaster.exe --blackBox true --bbSwaggerUrl https://petstore3.swagger.io/api/v3/openapi.json --bbTargetUrl https://petstore3.swagger.io --maxTime 30s --ratePerMinute 60 --outputFormat JAVA_JUNIT_5
```

# EvoMaster

```
* EvoMaster version: 3.1.0
* Loading configuration file from: C:\Users\arcu\WORK\code\EvoMaster\em.yaml
* WARNING: You are doing Black-Box testing, but you did not specify the 'problemType'. The system will default to RESTful API testing.
* Initializing...
10:31:47.901 [main] WARN o.e.c.problem.rest.param.BodyParam - Not supported data type: application/octet-stream
* There are 19 usable RESTful API endpoints defined in the schema configuration
10:31:47.955 [main] WARN o.e.c.s.gene.optional.ChoiceGene - cannot bind ChoiceGene with StringGene
* Starting to generate test cases
* Consumed search budget: 123.950%
* Covered targets: 44; time per test: 7427.2ms (7.6 actions); since last improvement: 8s
* Starting to apply minimization phase
* Recomputing full coverage for 5 tests
* Analyzing 5 tests with size greater than 1
* Minimization progress: 5/5
* Minimization phase took 76 seconds
* Evaluated tests: 5
* Evaluated actions: 38
* Needed budget: 100%
* Passed time (seconds): 114
* Execution time per test (ms): Avg=7427.20 , min=2995.00 , max=9006.00
* Execution time per action (ms): Avg=980.59 , min=904.11 , max=1000.67
* Computation overhead between tests (ms): Avg=15223.60 , min=4.00 , max=76086.00
* Going to save 21 tests to generated_tests
10:33:42.319 [main] WARN o.e.c.o.service.HttpWsTestCaseWriter - Currently no assertions are generated for response type: application/xml
10:33:42.323 [main] WARN o.e.c.o.service.HttpWsTestCaseWriter - Unhandled type for body payload: application/xml
* Potential faults: 13
* Successfully executed (HTTP code 2xx) 7 endpoints out of 20 (35%)
* EvoMaster process has completed successfully
* Use --help and visit http://www.evomaster.org to learn more about available options
```



```
@Test @Timeout(60)
public void test_1() throws Exception {

    given().accept("application/xml")
        .contentType("application/json")
        .body(" { " +
            " \"id\": 940, " +
            " \"name\": \"doggie\", " +
            " \"photoUrls\": [ " +
            " \"yHQXry\", " +
            " \"AZOgWb5y\", " +
            " \"GROBCmON\" " +
            " ], " +
            " \"tags\": [ " +
            " {}, " +
            " { " +
            " \"name\": \"nosupgc\" " +
            " } " +
            " ], " +
            " \"status\": \"pending\" " +
            " } ")
        .post(baseUrlOfSut + "/api/v3/pet")
        .then()
        .statusCode(200)
        .assertThat()
        .contentType("application/xml");
}
```

# Success Calls: Random but Valid Data

# Crashing with 500

```
@Test @Timeout(60)
public void test_4_with500() throws Exception {
    ExpectationHandler expectationHandler = expectationHandler();

    ValidatableResponse res_0 = given().accept("application/xml")
        .get(baseUrlOfSut + "/api/v3/user/8WIY1")
        .then()
        .statusCode(500)
        .assertThat()
        .contentType("application/xml");

    expectationHandler.expect(ems)
        .that(sco, Arrays.asList(200, 400, 404).contains(res_0.extract().statusCode()));
}
```

# Invalid response (eg status code not declared in schema)

```
@Test @Timeout(60)
public void test_8() throws Exception {
    ExpectationHandler expectationHandler = expectationHandler();

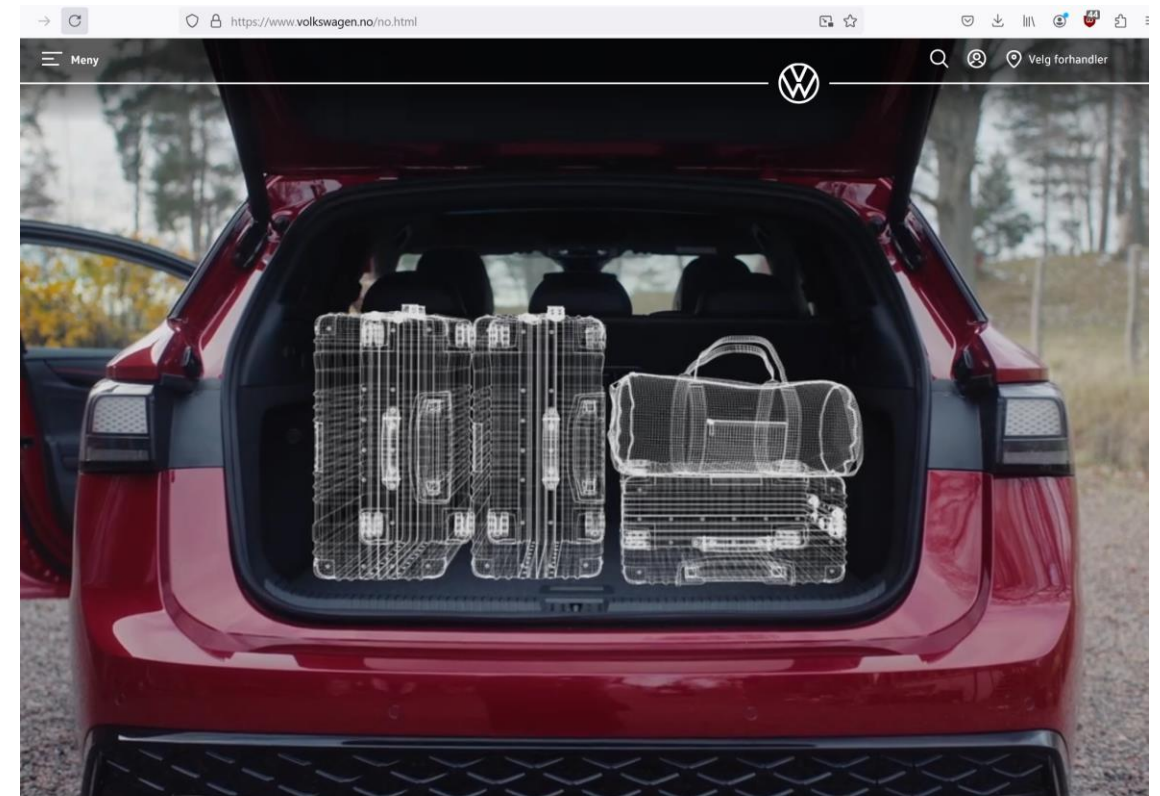
    ValidatableResponse res_0 = given().accept("application/json")
        .contentType("application/json")
        .body(" null ")
        .post(baseUrlOfSut + "/api/v3/store/order")
        .then()
        .statusCode(400)
        .assertThat()
        .contentType("application/json")
        .body(containsString("No Order provided. Try again?"));

    expectationHandler.expect(ems)
        .that(sco, Arrays.asList(200, 405).contains(res_0.extract().statusCode()));
}
```

# Experience With EvoMaster

- Author's of EvoMaster
- Academic tool, started in 2016
  - Around 30M NOK in funding from ERC and NFR
- Applied on many open-source APIs
  - found thousands of bugs
- Only tool supporting *white-box* testing
  - but only for JVM
- Academic collaborations with industry

# EvoMaster at Meituan and Volkswagen



# Challenges

- Lot of research in academia for better test generation strategies
- Cover larger parts of API code
- Find more faults (and fault types)
  - not all faults have same severity
- Test readability
  - testers still need to look at generated tests

Home > ACM Journals > ACM Transactions on Software Engineering and Methodology > Vol. 33, No. 1 > Testing RESTful APIs: A Survey

SURVEY | **OPEN ACCESS** | 



## Testing RESTful APIs: A Survey

Authors:  [Amid Golmohammadi](#),  [Man Zhang](#),  [Andrea Arcuri](#) | [Authors Info & Claims](#)

ACM Transactions on Software Engineering and Methodology, Volume 33, Issue 1 • Article No.: 27, Pages 1 - 41  
<https://doi.org/10.1145/3617175>

Published: 24 November 2023 [Publication History](#)



🗨️ 6 ↗️ 6,013



ACM Transactions on  
Software Engineering...

Volume 33, Issue 1

← Previous      Next →

### Abstract

- 1 Introduction
- 2 Background ✓
- 3 Related Work
- 4 Research Method ✓
- 5 Status of Publications in REST API Testing ✓
- 6 Approaches to Testing REST API ✓
- 7 Tools for Testing RESTful APIs ✓

### Abstract

In industry, RESTful APIs are widely used to build modern Cloud Applications. Testing them is challenging, because not only do they rely on network communications, but also they deal with external services like databases. Therefore, there has been a large amount of research sprout in recent years on how to automatically verify this kind of web services. In this article, we present a comprehensive review of the current state-of-the-art in testing RESTful APIs based on the analysis of 92 scientific articles. These articles were gathered by utilizing search queries formulated around the concept of RESTful API testing on seven popular databases. We eliminated irrelevant articles based on our predefined criteria and conducted a snowballing phase to minimize the possibility of missing any relevant paper. This survey categorizes and summarizes the existing scientific work on testing RESTful APIs and discusses the current challenges in the verification of RESTful APIs. This survey clearly shows an increasing interest among researchers in this field, from 2017 onward. However, there are still a lot of open research challenges to overcome.



# <https://github.com/WebFuzzing/EvoMaster/blob/master/docs/publications.md>

## Publications

The development of *EvoMaster* is rooted in academia. Here, you can find the PDFs of all the academic publications based on *EvoMaster*. Furthermore, slides of presentations can be found [here](#). These can be useful if you want to know more on how *EvoMaster* works internally, e.g., details on the Many Independent Objective (MIO) algorithm.

To help to replicate previous studies, for most of these papers we also provide the scripts used to setup the experiments. This explained in more details [here](#). Also, some of these papers provides full replication packages, which are linked directly in the papers (and not stored in this repository).

## Recent arXiv Technical Reports, not Peer-Reviewed (Yet)

- M. Zhang, A. Arcuri, Y. Li, K. Xue, Z. Wang, J. Huo, W. Huang. *Fuzzing Microservices In Industry: Experience of Applying EvoMaster at Meituan*. [\[arXiv\]](#)

## Peer-Reviewed Publications

### 2024

- S. Seran. *Search-based Security Testing of Enterprise Microservices*. IEEE International Conference on Software Testing, Validation and Verification (ICST), Doctoral Symposium. [\[PDF\]](#)
- A. Arcuri, M. Zhang, J.P. Galeotti. *Advanced White-Box Heuristics for Search-Based Fuzzing of REST APIs*. ACM Transactions on Software Engineering and Methodology (TOSEM). [\[PDF\]](#)[\[Scripts\]](#)

### 2023

- S. Seran, M. Zhang, A. Arcuri. *Search-Based Mock Generation of External Web Service Interactions*. Symposium on Search-based Software Engineering (SSBSE). [\[PDF\]](#)
- A. Golmohammadi, M. Zhang, A. Arcuri. *On the Impact of Tool Evolution and Case Study Size on SBSE Experiments: A Replicated Study with EvoMaster*. Symposium on Search-based Software Engineering (SSBSE). [\[PDF\]](#)
- A. Golmohammadi. *Enhancing White-Box Search-Based Testing of RESTful APIs*. IEEE International Symposium on

- If you want to go into the low-level details of how these techniques work



# Conclusion

- Many success stories about fuzzing
- REST fuzzing (and partially GraphQL and RPC) is getting momentum
- *Several open-source tools are available, to try out, today!*
  - we are biased about EvoMaster, but Schemathesis and Restler are good alternatives

# Q/A

Thanks!

On GitHub:

- WebFuzzing/EvoMaster
- microsoft/restler-fuzzer
- schemathesis/schemathesis
- Yelp/fuzz-lightyear
- isa-group/RESTest