Fuzz Testing Web APIs: Overview of Existing Tools

Prof. Andrea Arcuri

Kristiania University College and OsloMet

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, security issues)
  - **Maximize schema coverage**
  - **Maximize code coverage**
- 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

Fuzzers for REST APIs

- There are at least **25** open-source fuzzers out there for REST APIs
  - but many are just academic proof-of-concept
  - few have been discontinued (eg Dredd)

- Top 4 currently maintained fuzzers on **GitHub**
  - as of October 2025
- **Restler** (+2800⭐)
- **Schemathesis** (+2700⭐)
- **CATS** (+1300⭐)
- **EvoMaster** (+600⭐)

github.com/WebFuzzing/EvoMaster

WebFuzzing / EvoMaster

Type / to search

Code | Issues 40 | Pull requests 14 | Discussions | Actions | Projects | Wiki | Security 254 | Insights | Settings

EvoMaster

Public

Edit Pins | Unwatch 22 | Fork 96 | Starred 618

master | 231 Branches | 29 Tags | Go to file t | Add file | Code

arcuri82 Merge pull request #1327 from WebFuzzing/phg/dtoNonCustomT... 8cb90da · last week 12,489 Commits

.circleci	clarification	4 years ago
.github	upgraded dependency	3 weeks ago
.mvn	trying workaround for Kotlin compiler issue	2 years ago
client-java	modified debug logs to warn	3 weeks ago
core-driver-it	fixed 4.0.1-SNAPSHOT	last month
core-graphql-it	fixed 4.0.1-SNAPSHOT	last month
core-it	tests for findDeleteFor	last week
core	Merge branch 'master' into phg/dtoNonCustomTypes	last week
dbconstraint	fixed 4.0.1-SNAPSHOT	last month
docs	Debug and update	2 weeks ago
e2e-tests	Merge branch 'master' into phg/dtoNonCustomTypes	last week
outdated-client-dotnet	outdating no longer supported white-box testing of back...	last year
outdated-client-js	outdating no longer supported white-box testing of back...	last year
report	fixed 4.0.1-SNAPSHOT	last month
scripts	removed outdated files	last month
solver	fixed 4.0.1-SNAPSHOT	last month

About

The first open-source AI-driven tool for automatically generating system-level test cases (also known as fuzzing) for web/enterprise applications. Currently targeting whitebox and blackbox testing of Web APIs, like REST, GraphQL and RPC (e.g., gRPC and Thrift).

kotlin | java | testing | graphql | rest | grpc | thrift | evolutionary-algorithms | fuzzing | api-rest | fuzzer | api-testing | rpc-api | test-case-generation | search-based-software-testing

Readme
LGPL-3.0 license
Activity
Custom properties
618 stars
22 watching
96 forks
Report repository

Releases 22

v4.0.0 Latest
on Aug 11

+ 21 releases

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

https://petstore3.swagger.io/api/v3/openapi.json 240%

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON


```
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\nSwagger at [http://swagger.io](http://swagger.io). In the third iteration of the pet store, we've
                            switched to the design first approach!\nYou can now help us improve the API whether it's by making changes to
                            the definition itself or to the code.\nThat way, with time, we can improve the API in general, and expose
                            some of the new features in OAS3.\n\nSome useful links:\n- [The Pet Store repository](https://github.com/swag
                            ger-api/swagger-petstore)\n- [The source API definition for the Pet Store](https://github.com/swagger-api/swa
                            gger-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"
    version:                "1.0.19"
▼ externalDocs:
    description:            "Find out more about Swagger"
    url:                    "http://swagger.io"
▼ servers:
  ▼ 0:
      url:                  "/api/v3"
```

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 Python, Java, Kotlin and JavaScript

```
docker run
    -v "$(pwd)/generated_tests":/generated_tests
    webfuzzing/evomaster
    --blackBox true
    --maxTime 30s
    --ratePerMinute 60
    --bbSwaggerUrl  https://petstore.swagger.io/v2/swagger.json
```

example — -zsh — 135×50

```
Mon Sep 29 11:27:06 CEST 2025
arcuri82@Mac example % docker run -v "$(pwd)/generated_tests":/generated_tests webfuzzing/evomaster  --blackBox true --maxTime 30s  --r
atePerMinute 60 --bbSwaggerUrl  https://petstore.swagger.io/v2/swagger.json
WARNING: The requested image's platform (linux/amd64) does not match the detected host platform (linux/arm64/v8) and no specific platfo
rm was requested
*
 _____          ___  ___          _
|  ___|         |  \/  |         | |
| |____   _____ | .  . | __ _ ___| |_ ___ _ __
|  __\ \ / / _ \| |\/| |/ _` / __| __/ _ \ '__|
| |___\ V / (_) | |  | | (_| \__ \ ||  __/ |
\____/ \_/ \___/\_|  |_/\__,_|___/\__\___|_|


* EvoMaster version: 4.0.0
* WARNING: You are doing Black-Box testing, but you did not specify the 'problemType'. The system will default to RESTful API testing.
* WARNING: You are doing Black-Box testing, but you did not specify the 'outputFormat'. The system will default to PYTHON_UNITTEST.
* Going to create configuration file at: /em.yaml
* Loading configuration file from: /em.yaml
* You are running EvoMaster inside Docker. To access the generated test suite under '/generated_tests', you will need to mount a folder
 or volume. Also references to host machine on 'localhost' would need to be replaced with 'host.docker.internal'. If this is the first
time you run EvoMaster in Docker, you are strongly recommended to first check the documentation at: https://github.com/WebFuzzing/EvoMa
ster/blob/master/docs/docker.md
* Initializing...
* There are 20 usable RESTful API endpoints defined in the schema configuration
* There are 2 detected issues when analyzing the schema. These are not necessarily problems in the schema, but possible (temporary) lim
itations of EvoMaster itself.
* 0: Not supported content types for body payload in POST:/v2/pet/{petId}/uploadImage : multipart/form-data
* 1: The use of 'example' inside a Schema Object is deprecated in OpenAPI. Rather use 'examples'. Read value: doggie
* Starting to generate test cases
* Consumed search budget: 104.193%
* Covered targets: 104; time per test: 1354.9ms (1.1 actions); since last improvement: 7s
* Starting to apply minimization phase
* Recomputing full coverage for 20 tests
* No test to minimize
* Minimization phase took 25 seconds
* Evaluated tests: 23
* Evaluated actions: 25
* Needed budget: 80%
* Passed time (seconds): 57
* Execution time per test (ms): Avg=1354.91 , min=950.00 , max=3002.00
* Execution time per action (ms): Avg=1246.78 , min=950.00 , max=2004.00
* Computation overhead between tests (ms): Avg=1094.26 , min=0.00 , max=25076.00
* Starting to apply security testing
* Going to save 21 tests to generated_tests
* Potential faults: 14
* Successfully executed (HTTP code 2xx) 13 endpoints out of 21 (62%)
* EvoMaster process has completed successfully
* Use --help and visit https://www.evomaster.org to learn more about available options
arcuri82@Mac example %
```

Success Calls: Random but Valid Data

```
# Calls:
# (200) GET:/v2/pet/findByTags
@timeout_decorator.timeout(60)
def test_1_get_on_findByTags_returns_empty_list(self):

    headers = {}
    headers['Accept'] = "application/json"
    res_0 = requests \
            .get(self.baseUrlOfSut + "/v2/pet/findByTags?tags=lPADYDnRLOwnjsdW&tags=chS0o&tags=Vff5S_j7W&tags=Ps",
                headers=headers, timeout=60)

    assert res_0.status_code == 200
    assert "application/json" in res_0.headers["content-type"]
    assert len(res_0.json()) == 0
```

Schema Mismatch (eg undeclared 200)

```python
# Calls:
# (200) PUT:/v2/user/{username}
# Found 1 potential fault of type-code 101
@timeout_decorator.timeout(60)
def test_8_put_on_user_returnsMismatchResponseWithSchema(self):

    # Fault101. Received A Response From API With A Structure/Data That I
    headers = {}
    headers["content-type"] = "application/json"
    body = {}
    body = " { " + \
        " \"firstName\": \"t3PeK1x\", " + \
        " \"lastName\": \"1x_eQMjnWztpWGj\", " + \
        " \"email\": \"c0xQmHfJJU4OjPXp\", " + \
        " \"phone\": \"vSlgsZ\", " + \
        " \"userStatus\": 649 " + \
        " } "
    headers['Accept'] = "*/*"
    res_0 = requests \
            .put(self.baseUrlOfSut + "/v2/user/VDJDKy",
                headers=headers, data=body, timeout=60)

    assert res_0.status_code == 200
    assert "application/json" in res_0.headers["content-type"]
    assert res_0.json()["code"] == 200.0
    assert res_0.json()["type"] == "unknown"
    assert res_0.json()["message"] == "0"

    # Cleanup actions
    headers = {}
    headers['Accept'] = "*/*"
```

petstore.swagger.io/v2/swagger.json

JSON | Raw Data | Headers

Save | Copy | Collapse All | Expand All | Filter JSON

```
▸ /store/order/{orderId}:        { get: {…}, delete: {…} }
▸ /user/createWithList:          { post: {…} }
▾ /user/{username}:
  ▸ get:                         { summary: "Get user by user name", operationId: "
  ▾ put:
    ▾ tags:
        0:                       "user"
      summary:                   "Updated user"
      description:               "This can only be done by the logged in user."
      operationId:               "updateUser"
    ▾ consumes:
        0:                       "application/json"
    ▾ produces:
        0:                       "application/json"
        1:                       "application/xml"
    ▾ parameters:
      ▾ 0:
          name:                  "username"
          in:                    "path"
          description:           "name that need to be updated"
          required:              true
          type:                  "string"
      ▾ 1:
          in:                    "body"
          name:                  "body"
          description:           "Updated user object"
          required:              true
        ▾ schema:
            $ref:                "#/definitions/User"
    ▾ responses:
      ▾ 400:
          description:           "Invalid user supplied"
      ▾ 404:
          description:           "User not found"
```

Interactive Test Reports

http://localhost:8000 80%

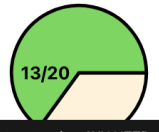
WEB FUZZING COMMONS

Creation Date: Mon, 29 Sep 2025 09:28:13 GMT **Tool: EvoMaster-4.0.0** **Schema Version: 0.0.1**

Overview | Endpoints

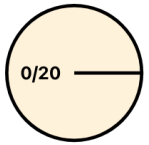
REST Report

2XX

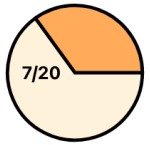


Test cases cover 7 endpoints returning 4XX HTTP codes out of 20 endpoints.

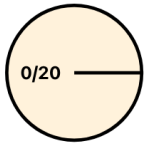
3XX



4XX



5XX



 # Endpoints: **20**

 # Generated Test Files: **3**
 # Generated Tests Cases: **21**
 # HTTP Calls: **21**

Test Files

- EvoMaster_faults_Test.py (# 14)
- EvoMaster_successes_Test.py (# 4)
- EvoMaster_others_Test.py (# 3)

Total Faults: **14**
Distinct Fault Types: **1**

Codes	Name	Ratio	#
101	Received A Response From API With A Structure/Data That Is Not Matching Its Schema	13/20	13

http://localhost:8000 80%

WEB FUZZING COMMONS

Creation Date: Mon, 29 Sep 2025 09:28:13 GMT **Tool: EvoMaster-4.0.0** **Schema Version: 0.0.1**

Overview | Endpoints

Filter by HTTP Status Code

H200 H404 H415

Filter by Fault Code

F101

Click to toggle: Default → Active → Removed → Default

Code Documentation

Endpoints: **20** / 20

- POST:/v2/pet/{petId}/uploadImage — H415 F101
- POST:/v2/pet — H200 F101
- PUT:/v2/pet — H200 F101
- GET:/v2/pet/findByStatus — H200
- GET:/v2/pet/findByTags — H200

HTTP CODES: H200

FAULT CODES: *No faults recorded for this endpoint.*

Click to show test cases.

EvoMaster_successes_Test.py#test_1_get_on_findByTags_returns_empty_list 200

- GET:/v2/pet/{petId} — H404 F101
- POST:/v2/pet/{petId} — H404 F101
- DELETE:/v2/pet/{petId} — H404

Experience With EvoMaster

- Author’s of EvoMaster
- Academic tool, started in 2016
  - Around 3 millions in funding from ERC and NFR
- Applied on many open-source APIs
  - found thousands of bugs
- Only tool supporting *white-box* testing 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

Hmmmm… why not just using a LLM?

- Input: OpenAPI schema
- Output: test cases

- Can work, but poor results
- You would miss all information from the responses of API
- No way to tell if a test case has found a fault
- You must interact with the API

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
- Endava/cats