

Search-Based System Testing with EvoMaster

Prof. Andrea Arcuri

Kristiania University College, Norway

Oslo Metropolitan University, Norway

In this tutorial

1. Brief intro REST web services
2. Brief overview of the *EvoMaster* tool
3. Demo / Walkthrough

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

RESTful APIs

- Most common type of web services
 - others are *SOAP* and *GraphQL*
- Access of set of resources using HTTP
- REST is not a protocol, but just architectural guidelines on how to define HTTP endpoints
 - hierarchical URLs to represent resources
 - HTTP verbs (GET, POST, PUT, DELETE, etc.) as “actions” on resources

Resource Summary

- Files
- About
- Changes
- Children
- Parents
- Permissions
- Revisions
- Apps
- Comments
- Replies
- Properties
- Channels
- Realtime
- Teamdrives
- Standard Features

API Reference



This API reference is organized by resource type. Each resource type has one or more data representations and one or more methods.

Resource types

Files

For Files Resource details, see the [resource representation](#) page.

Method	HTTP request	Description
URIs relative to https://www.googleapis.com/drive/v2 , unless otherwise noted		
get	GET <code>/files/<i>fileId</i></code>	Gets a file's metadata by ID.
insert	POST <code>https://www.googleapis.com/upload/drive/v2/files</code> and POST <code>/files</code>	Insert a new file.
patch	PATCH <code>/files/<i>fileId</i></code>	Updates file metadata. This method supports patch semantics .
update	PUT <code>https://www.googleapis.com/upload/drive/v2/files/<i>fileId</i></code> and PUT <code>/files/<i>fileId</i></code>	Updates file metadata and/or content.
copy	POST <code>/files/<i>fileId</i>/copy</code>	Creates a copy of the specified file.

Getting started with the REST API

The foundation of all digital integrations with LinkedIn

The REST API is the heart of all programatic interactions with LinkedIn. All other methods of interacting, such as the JavaScript and Mobile SDKs, are simply wrappers around the REST API to provide an added level of convenience for developers. As a result, even if you are doing mobile or JavaScript development, it's still worth taking the time to familiarize yourself with how the REST API works and what it can do for you.



reddit

API DOCUMENTATION

arcuri82 (1) | | [preferences](#) | [logout](#)

API methods

by section

by oauth scope

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

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.

Twitter Developer Documentation

[Docs](#) / [REST APIs](#)

Products & Services

[Best practices](#)[API overview](#)[Twitter for Websites](#)[Twitter Kit](#)[Cards](#)[OAuth](#)[REST APIs](#)[API Rate Limits](#)[Rate Limits: Chart](#)[The Search API](#)[The Search API: Tweets by Place](#)

REST APIs

The [REST APIs](#) provide programmatic access to read and write Twitter data. Create a new Tweet, read user profile and follower data, and more. The REST API identifies Twitter applications and users using [OAuth](#); responses are in JSON format.

If your intention is to monitor or process Tweets in real-time, consider using the [Streaming API](#) instead.

Overview

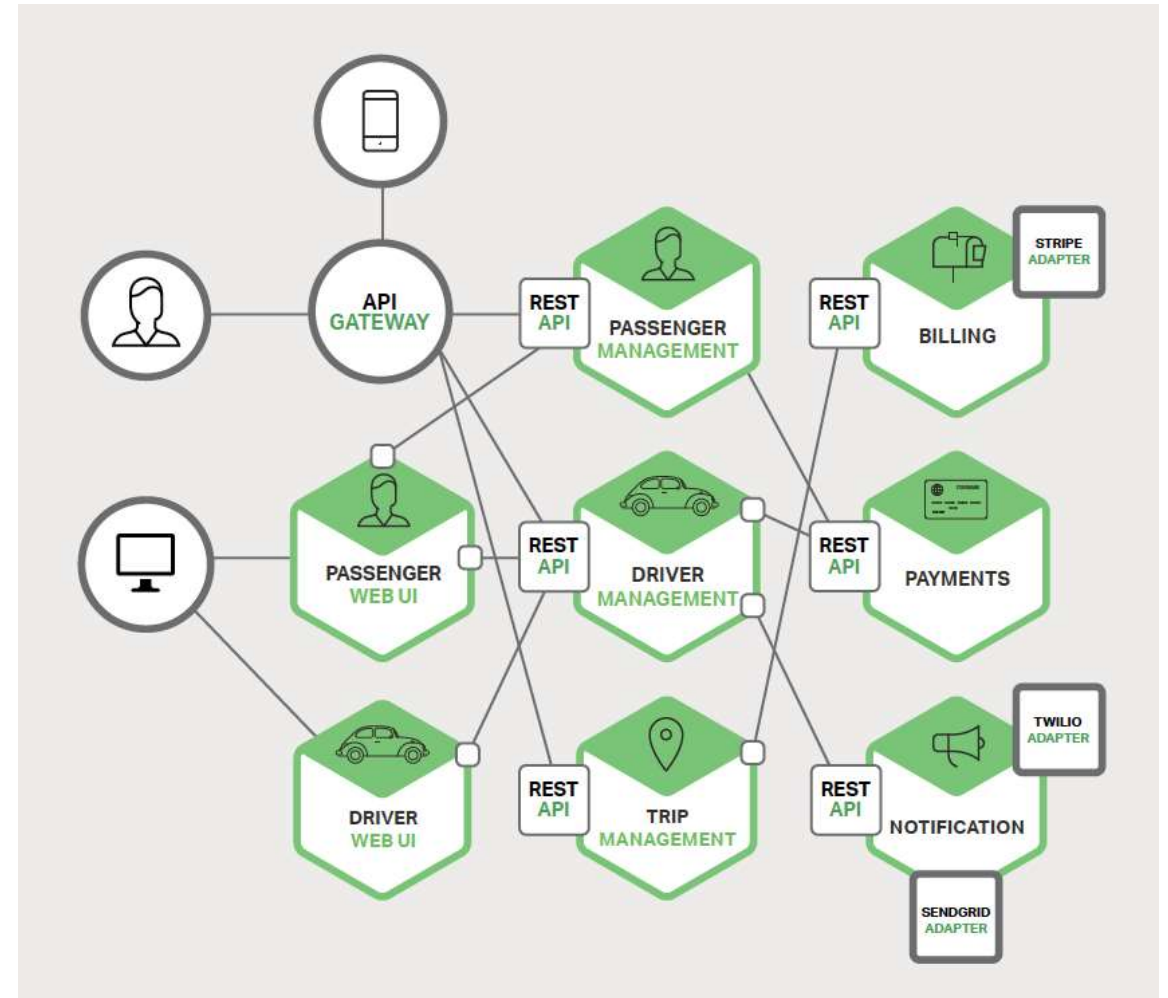
Below are some documents that will help you get going with the REST APIs as quickly as possible

- [API Rate Limiting](#)
- [API Rate Limits](#)
- [Working with Timelines](#)
- [Using the Twitter Search API](#)
- [Finding Tweets about Places](#)
- [Uploading Media](#)
- [Reference Documentation](#)

Default entities and retweets

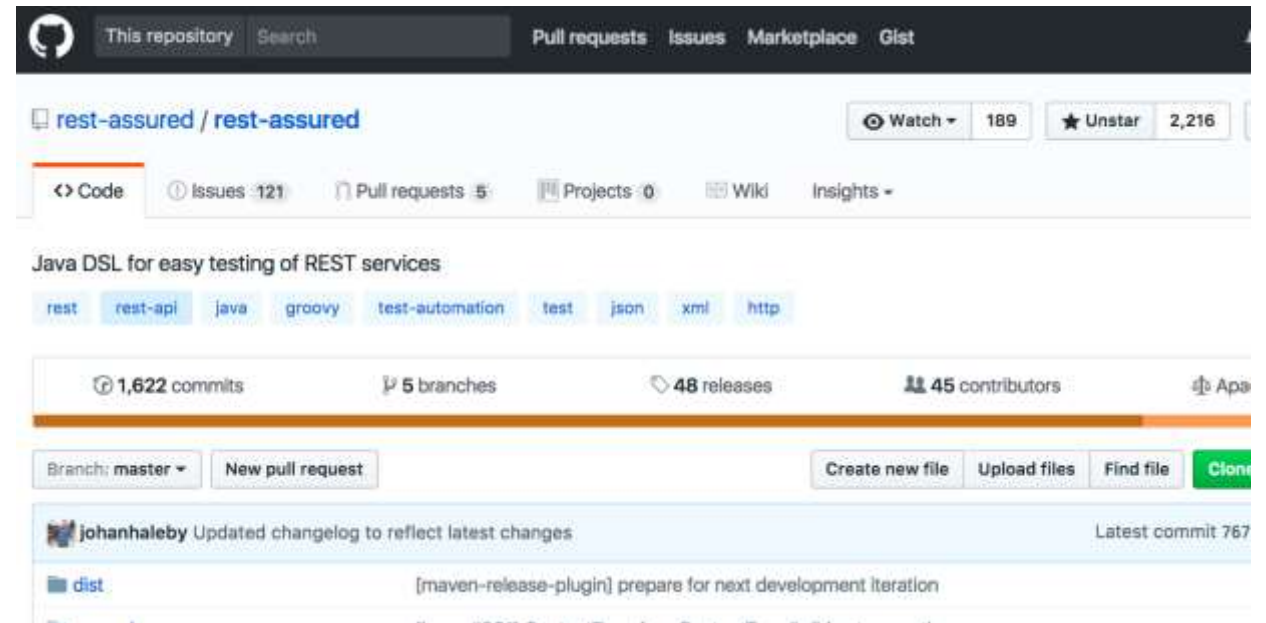
REST in Microservices

- Common trend in enterprises
 - Amazon, Netflix, etc.
- Split application in many small web services, typically REST
- Easier to scale and maintain
- User (browser/app) has no idea on how backed is architected, only see one entry point, eg “API Gateway”



Testing of REST APIs

- Do HTTP calls, read responses
- Setup database states
- Specialized libraries, eg in Java the popular **RestAssured**



@Test

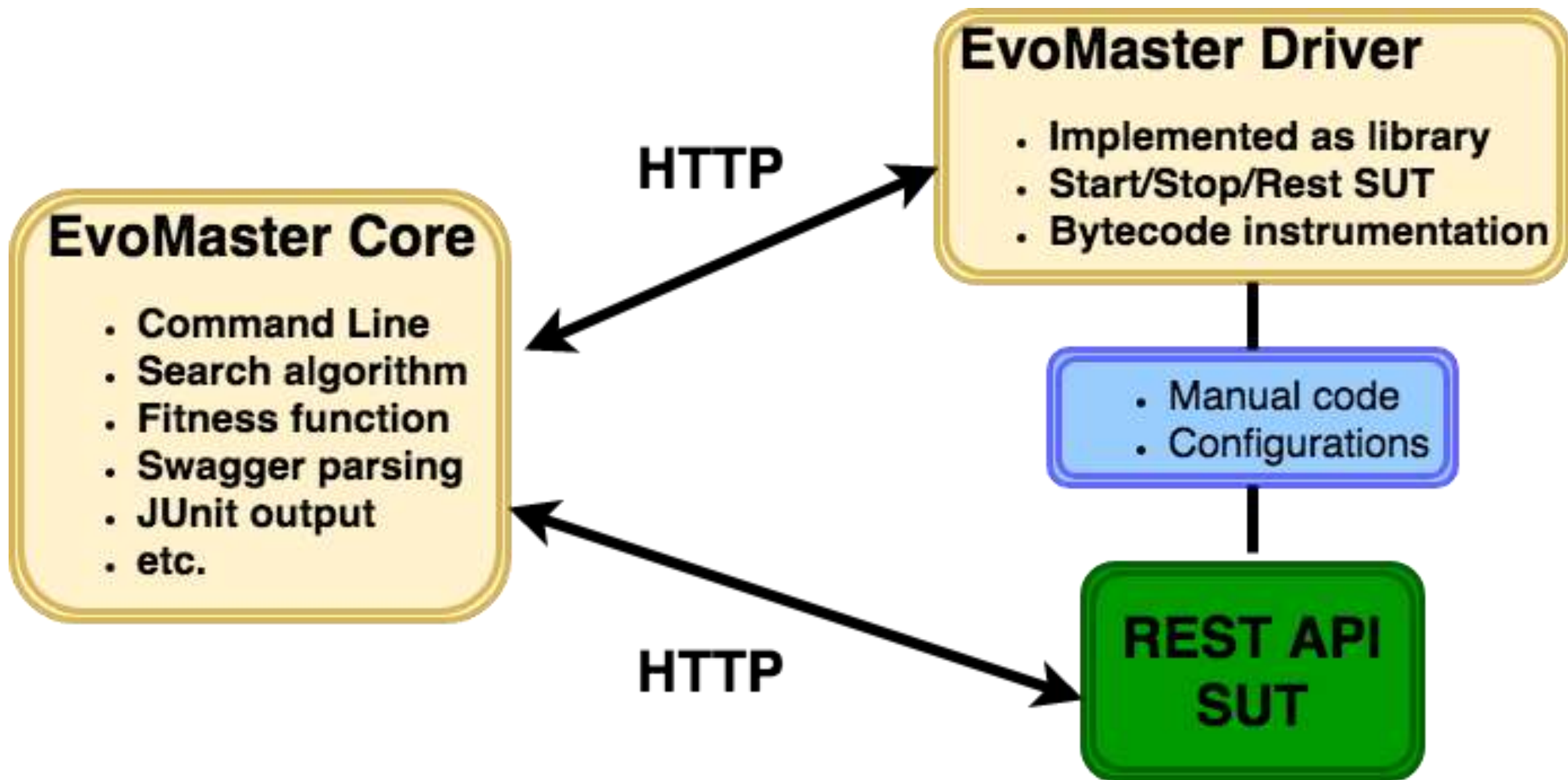
public void test0() throws Exception {

```
    given().header("Authorization", "ApiKey user")
        .accept("*/*")
        .get("www.foo.com/api/v1/media_files/42")
        .then()
        .statusCode(200);
```

}

EvoMaster

- Tool to automatically generate tests for REST APIs
- **White box**
 - can exploit structural and runtime information of the SUT
 - advanced *testability transformations*
 - SQL handling at runtime
 - Etc.
- Search-based testing technique (**SBST**)
 - MIO Algorithm + Adaptive Hypermutation
- Fully automated
- Open-source prototype: www.evomaster.org
- Currently targeting JVM languages (eg Java and Kotlin)



www.evomaster.org

The screenshot shows the GitHub repository page for `EMResearch/EvoMaster`. The repository is public and has 18 watchers, 191 stars, and 37 forks. The main navigation bar includes links for Code, Issues (9), Pull requests (8), Discussions, Actions, Projects, Wiki, Security (1), Insights, and Settings. The repository is currently on the `master` branch, with 32 branches and 10 tags. A recent commit by `arcuri82` is highlighted, showing a merge of pull request #392 from `EMResearch/exp-script`. The commit message is "Merge pull request #392 from EMResearch/exp-script" and it includes a list of files and their changes. The right sidebar contains the "About" section, which describes the project as the first open-source AI-driven tool for automatically generating system-level test cases (also known as fuzzing) for web/enterprise applications. It also lists tags for `kotlin`, `java`, `testing`, `rest`, `evolutionary-algorithms`, `api-rest`, `api-testing`, and `test-case-generation`. Below the "About" section, there is a "Releases" section showing the latest release, `v1.2.1`, which was released on July 1.

https://github.com/EMResearch/EvoMaster

Search or jump to... Pull requests Issues Marketplace Explore

EMResearch / EvoMaster Public

Unwatch 18 Unstar 191 Fork 37

<> Code Issues 9 Pull requests 8 Discussions Actions Projects Wiki Security 1 Insights Settings

master 32 branches 10 tags

Go to file Add file Code

arcuri82 Merge pull request #392 from EMResearch/exp-script ✓ b1a32db 5 days ago 4,118 commits

.circleci	clarification	last month
.github/workflows	first E2E for C#	3 months ago
client-dotnet/src	warning.html as embedded resource	3 months ago
client-java	update input check for cleanDataInTables	12 days ago
client-js	handle replaced ternary with await	last month
core-driver-it	refactoring package name	last month
core-graphql-it	1.2.2-SNAPSHOT	3 months ago
core-it	fix a bug that uses the mismatched inputs	3 months ago
core	Merge pull request #388 from EMResearch/tmp-handle-1nd1	9 days ago
dbconstraint	support enum and handle the constraints in mysql	3 months ago
docs	Merge pull request #392 from EMResearch/exp-script	5 days ago
e2e-tests	Merge pull request #388 from EMResearch/tmp-handle-1nd1	9 days ago

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 REST APIs.

kotlin java testing rest evolutionary-algorithms api-rest api-testing test-case-generation

Readme

LGPL-3.0 License

Releases 10

v1.2.1 Latest on Jul 1

+ 9 releases

Demo