

AGORA Revisited: A Replication Study on Innovative Test Oracle Generation for Advancing the Robustness of REST APIs

PROJECT PLAN PRESENTATION

Presentation By Parth Shah, Sankaranarayanan Srinivasan, Harshini Reddy Koukuntla, Taruni Movva
February 14, 2024

The Research Questions

This replication study aims to address the following primary research questions:

- a. How does the performance of AGORA scale with larger datasets of API requests and responses? Are there any bottlenecks in the system as the dataset size increases?
- b. How does detecting and reducing duplicate input requests affect the performance of AGORA's API testing process?
- c. How do different configurations of JSONMutator mutation operators affect the failure detection ratio achieved by AGORA?

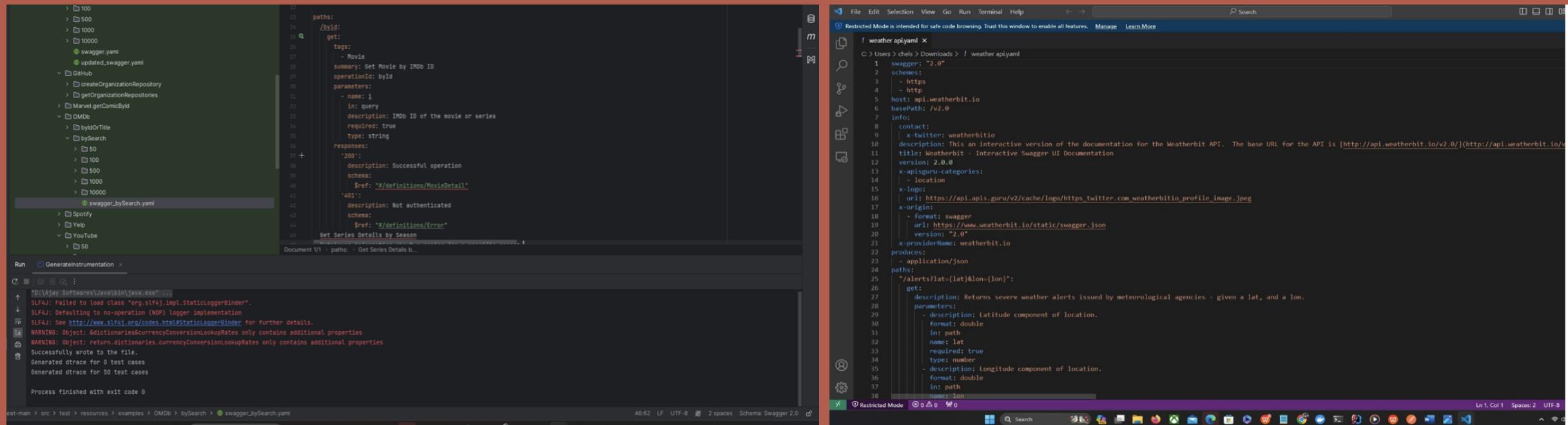
RQ - 1 : Evaluating AGORA's Performance with Larger Datasets

Approach Overview:

Developed two primary approaches to assess AGORA's scalability with enhanced datasets

Approach 1: Expanded OMDB dataset with new sources like "Get Series Details by Season."

Approach 2: Integrated Weather API obtained from APIs.guru for numerical data.



The image shows a dual-pane interface. On the left, a terminal window displays Java compilation logs, including error messages about failed class loading and static logger binder issues. On the right, a code editor displays an OpenAPI specification (YAML) for a Weather API. The YAML file includes definitions for schemes, host, basePath, info, contact, and paths, specifically for an 'alerts' endpoint that takes latitude and longitude parameters.

```
weather api.yaml
swagger: "2.0"
schemes:
  - https
  - http
host: api.weatherbit.io
basePath: /v2.0
info:
  contact:
    x-twitter: weatherbitio
  description: This is an interactive version of the documentation for the Weatherbit API. The base URL for the API is http://api.weatherbit.io/v2.0
  title: Weatherbit - Interactive Swagger UI Documentation
  version: 2.0.0
xapisguru-categories:
  - location
x-logo:
  url: https://api.apis.guru/v2/cache/logo/https://twitter.com/weatherbitio_profile_image.jpeg
x-origin:
  - format: swagger
  - url: https://www.weatherbit.io/static/swagger.json
  - version: "2.0"
x-providerName: weatherbit.io
produces:
  - application/json
paths:
  "/alerts?lat=[lat]&lon=[lon]":
    get:
      description: Returns severe weather alerts issued by meteorological agencies - given a lat, and a lon.
      parameters:
        - description: Latitude component of location.
          format: double
          in: path
          name: lat
          required: true
        - description: Longitude component of location.
          format: double
          in: path
          name: lon
```

One of the sources incorporated to expand the OMDB dataset

OpenAPI Spec for the Weather API

Data Expansion:

- Expanded OMDB dataset to include diverse sources suitable for various data types.
- Integration of Weather API to evaluate AGORA's ability in handling arithmetic comparisons.

Analysis Process:

- Generated API responses for endpoints, ensuring adherence to OpenAPI format.
- Organized responses into folders by dataset size using POSTMAN tool.
- Utilized BEET and IntelliJ IDE for creating .decl and .dtrace files, and Docker for Daikon execution.

Here is the example of the generated invariant for the weather API Response

- 1) /weather-api&getCurrentWeather&200&data():::EXIT;data.attributes.temp is Numeric;daikon.inv.unary.string.IsNumeric;(data.attributes.temp)
- 2) /weather-api&getCurrentWeather&200&data():::EXIT;data.attributes.weather one of {"Mild", "Sweltering", "Scorching", "Frost", "Cool", "Humid", "Rain", "Partly Cloudy", "Freezing", "Extreme Heat", "Muggy", "Chilly", "Breezy", "Clear", "Cloudy", "Wet"};daikon.inv.unary.scalar.OneOfScalar;(data.attributes.weather)
- 3) /weather-api&getCurrentWeather&200&data():::EXIT;data.type == "weather";daikon.inv.binary.twoString.StringEqual;(data.type)

snapshot of the generated endpoint

```
testCasedId,queryParameters,operationId,path,httpMethod,headerParameters,status_code,passed,failReason,responseBody
a3965e08-0399-4fc6-8f5c-31a9c93c2ac,postal_code=100-0001,getWeatherOrAirQualityData,/current/airquality?postal_code={postal_code},GET,Authorization-Bearer+randomAccessToken,200,
2152205e-69c8-478e-b56e-3bb9c95099919,city=tokyo;country=japan,getWeatherOrAirQualityData,/current/airquality?city={city}&country={country},GET,Authorization-Bearer+randomAccessToken
39eeaa8f-a209-4fea-9fc9-9e4baa6984,postal_code=100-0001,getWeatherOrAirQualityData,/current/airquality?postal_code={postal_code},GET,Authorization-Bearer+randomAccessToken,200,
a5314703-9685-4aeb-9e18-13caa634abf1,city_id=12345,getWeatherOrAirQualityData,/current?city_id={city_id},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none,{"data": [{"typ
cebef332-4ef0-4e9d-8c89-31255f438fe6","cities":12345,67890},getWeatherOrAirQualityData,/current?cities={cities},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none},{"data": [
4e150b8e-eed7-423e-834d-7d152c9f71,city=tokyo;country=japan,getWeatherOrAirQualityData,/current/airquality?city={city}&country={country},GET,Authorization-Bearer+randomAccessToken
2dd58f71-a15b-42c4-a212-7ff6fdc97ac,lat=35.6895;lon=139.6917,getWeatherOrAirQualityData,/current/airquality?lat={lat}&lon={lon},GET,Authorization-Bearer+randomAccessToken,200,
8b4d165f-0281-4701-bdaa-82cae7a3ee,lat=35.6895;lon=139.6917,getWeatherOrAirQualityData,/current/airquality?lat={lat}&lon={lon},GET,Authorization-Bearer+randomAccessToken,200,TR
1a7db91e-d00e-47b1-86ff-8e4edbd84c8c,city=tokyo;country=japan,getWeatherOrAirQualityData,/current?city={city}&country={country},GET,Authorization-Bearer+randomAccessToken,200,TR
f08a0e0d-b050-41db-8934-5e6f44524f52,city_id=12345,getWeatherOrAirQualityData,/current/airquality?city_id={city_id},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none,>{"da
43e254ab-deef-4bb5-b2d5-3429e32c174d,city_id=12345,getWeatherOrAirQualityData,/current/airquality?city_id={city_id},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"da
ee57f81b-59a5-4d8c-b4be-9c36d83872e,city=tokyo;country=japan,getWeatherOrAirQualityData,/current?city={city}&country={country},GET,Authorization-Bearer+randomAccessToken,200,TR
bc6cd0fc-373e-46cb-854f-03cd7fb0d46,lat=35.6895;lon=139.6917,getWeatherOrAirQualityData,/current/airquality?lat={lat}&lon={lon},GET,Authorization-Bearer+randomAccessToken,200,TR
bf23446b-90b1-42a4-8030-56e7ba1610c,city_id=12345,67890},getWeatherOrAirQualityData,/current/airquality?city_id={city_id},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"da
59a0fffd-15ad-4570-9af0-ba280f25385,city_id=12345,getWeatherOrAirQualityData,/current/airquality?city_id={city_id},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"da
e352ab9d-6bfc-4baf-8990-8515a04265e4,city_id=12345,getWeatherOrAirQualityData,/current/airquality?city_id={city_id},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"da
b88c179-55e6-4f7e-4f7e-4d026a114842,city_id=12345,getWeatherOrAirQualityData,/current/airquality?city_id={city_id},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"da
c9a7fe45-8d43-4085-4b64-995c28961ecf,"cities":12345,67890},getWeatherOrAirQualityData,/current?cities={cities},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"data": [
ffb9392b-abc9-4ba2-8bd9-fdb1658a1b62,city=tokyo;country=japan,getWeatherOrAirQualityData,/current/airquality?city={city}&country={country},GET,Authorization-Bearer+randomAccessToken
e3a628c8-01a2-4723-9776-0c9fa93c268,"cities":12345,67890},getWeatherOrAirQualityData,/current?cities={cities},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"data": [
093ca557-3252-4994-b2d7-f0cf24898174,lat=35.6895;lon=139.6917,getWeatherOrAirQualityData,/alerts?lat={lat}&lon={lon},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"d
10a9e663-ac74-47e0-8815-abcf5598a91,postal_code=100-0001,getWeatherOrAirQualityData,/current/airquality?postal_code={postal_code},GET,Authorization-Bearer+randomAccessToken,200,
0eaabfb3-8c35-4642-b30c-4e8292c73f1,lat=35.6895;lon=139.6917,getWeatherOrAirQualityData,/current/airquality?lat={lat}&lon={lon},GET,Authorization-Bearer+randomAccessToken,200,
b2521a3f-482c-432b-98fa-4ee0ed24882,lat=35.6895;lon=139.6917,getWeatherOrAirQualityData,/alerts?lat={lat}&lon={lon},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"d
2344178a-d9c0-44d2-aacc-63f2a885e06b,city_id=12345,67890},getWeatherOrAirQualityData,/current?cities={cities},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"data": [
7968fd9b0c-c12a-4735-9314-40d5feaf9f860,postal_code=100-0001,getWeatherOrAirQualityData,/current/airquality?postal_code={postal_code},GET,Authorization-Bearer+randomAccessToken,200,
2dc612a9-746d-439e-86eb-941a2c745d66,lat=35.6895;lon=139.6917,getWeatherOrAirQualityData,/current/airquality?lat={lat}&lon={lon},GET,Authorization-Bearer+randomAccessToken,200,TR
61a917c7-bbd8-4a13-91d9-de2ac0e5254,city_id=12345,67890},getWeatherOrAirQualityData,/current?city_id={city_id},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"data": [
b353b381-f39b-4321-95ab-9ae943681f,lat=35.6895;lon=139.6917,getWeatherOrAirQualityData,/current/airquality?lat={lat}&lon={lon},GET,Authorization-Bearer+randomAccessToken,200,
5b67f13e-8eb1-485c-bd0d-0687f5a170f1,city=tokyo;country=japan,getWeatherOrAirQualityData,/current?city={city}&country={country},GET,Authorization-Bearer+randomAccessToken,200,TR
958ed64c-5c51-4563-908c-054e09d83d14,city=tokyo;country=japan,getWeatherOrAirQualityData,/current/airquality?city={city}&country={country},GET,Authorization-Bearer+randomAccessToken
767eb59e-5dab-4a60-9537-0f15ba22395f,lat=35.6895;lon=139.6917,getWeatherOrAirQualityData,/alerts?lat={lat}&lon={lon},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"d
4b5ceb88-fd5a-4ea7-a4f1-c1fc1e88a5,"cities":12345,67890},getWeatherOrAirQualityData,/current?cities={cities},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"data": [
cc197c5f-98eb-4a12-ae87-1f1fe3ba70eb,city=tokyo;country=japan,getWeatherOrAirQualityData,/current?city={city}&country={country},GET,Authorization-Bearer+randomAccessToken,200,TR
b2778581-bb07-4d8b-801-185e37d276e1,"cities":12345,67890},getWeatherOrAirQualityData,/current?city_id={city_id},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"data": [
08cf11a9-3f2f-4273-abdc-ab93c8b728f,city_id=12345,67890},getWeatherOrAirQualityData,/current?city_id={city_id},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"data": [
dfaaaf69-d835-4d7a-89ca-9a4d988607b77,city_id=12345,67890},getWeatherOrAirQualityData,/current?city_id={city_id},GET,Authorization-Bearer+randomAccessToken,200,TRUE,none, {"data": [
01f1c011-spaces:4-utf-8-crf
]
```

Primary objective:

- Assess AGORA's performance using enhanced datasets.
- AGORA employs Dynamic Program Analysis, effectiveness influenced by data quality and quantity.
- Confidence in AGORA's efficiency improvement as dataset size expands.

Efficiency Measurement:

- Efficiency formula: Efficiency = Number of relevant invariants generated from the extended dataset / Total number of invariants generated from the authors' dataset.
- Emphasize the importance of measuring AGORA's efficiency for larger datasets compared to the existing dataset.

RQ - 2 : Impact of Duplicate Requests reduction on AGORA's performance

- Our initial analysis of the AGORA dataset revealed a significant presence of duplicate input requests.

- We observed that the existing agora dataset is utilizing duplicate inputs (i.e., API Request endpoints) for Beet to evaluate invariants using the modified version of Daikon.

- We identified that the "createPlaylist" endpoint had duplicate api requests with the identical 'userid' parameter. In a similar vein, the "getAlbumTrack" endpoint was called with the 'market' parameter consistently across requests.

- The redundancy of these inputs can potentially skew the performance metrics of the API testing process, leading to inefficiency in both testing efforts and resource utilization.

Example – Spotify REST API

createPlaylist

operationId	path	httpMethod
createPlaylist	/users/{user_id}/playlists	POST

getAlbumTracks

F	G	H	I
operationId	path	httpMethod	
market=AD;offset=7;limit=4;	getAlbumTracks	/albums/{id}/tracks	GET
market=ID;	getAlbumTracks	/albums/{id}/tracks	GET
market=IE;	getAlbumTracks	/albums/{id}/tracks	GET
market=IE;	getAlbumTracks	/albums/{id}/tracks	GET
market=IE;offset=12;limit=38;	getAlbumTracks	/albums/{id}/tracks	GET
market=IL;	getAlbumTracks	/albums/{id}/tracks	GET
market=IN;	getAlbumTracks	/albums/{id}/tracks	GET
market=IN;	getAlbumTracks	/albums/{id}/tracks	GET
market=IN;limit=12;	getAlbumTracks	/albums/{id}/tracks	GET
market=IN;limit=38;	getAlbumTracks	/albums/{id}/tracks	GET
market=IN;offset=10;	getAlbumTracks	/albums/{id}/tracks	GET
market=IT;	getAlbumTracks	/albums/{id}/tracks	GET
market=IT;limit=32;	getAlbumTracks	/albums/{id}/tracks	GET
market=JO;	getAlbumTracks	/albums/{id}/tracks	GET
market=KE;offset=19;	getAlbumTracks	/albums/{id}/tracks	GET
market=KE;offset=19;	getAlbumTracks	/albums/{id}/tracks	GET
market=KE;offset=8;	getAlbumTracks	/albums/{id}/tracks	GET
market=KE;offset=8;	getAlbumTracks	/albums/{id}/tracks	GET
market=KG;offset=8;	getAlbumTracks	/albums/{id}/tracks	GET
market=KH;offset=3;	getAlbumTracks	/albums/{id}/tracks	GET
market=KI;offset=12;limit=47;	getAlbumTracks	/albums/{id}/tracks	GET
market=KM;	getAlbumTracks	/albums/{id}/tracks	GET
market=KM;	getAlbumTracks	/albums/{id}/tracks	GET
market=KN;offset=12;limit=46;	getAlbumTracks	/albums/{id}/tracks	GET

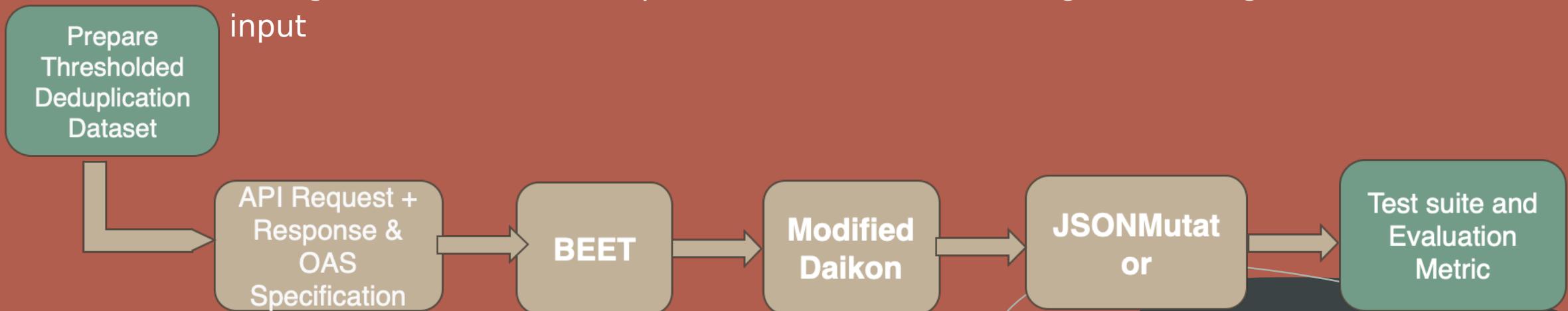
Plan

Experiment 1

- We will set a threshold value to control duplicate REST API requests in industrial REST API datasets. We will keep duplicate records up to threshold value and give dataset csv and OAS Specification to Beet as an input. After modified daikon will use result of beet to produce invariants.
- We will run JSONMutator tool to generate the test suite for new dataset which has duplicate API requests (duplicates limited to threshold).
We will detect if there are new invariants generated and what is the impact on test suite.

Expected output:

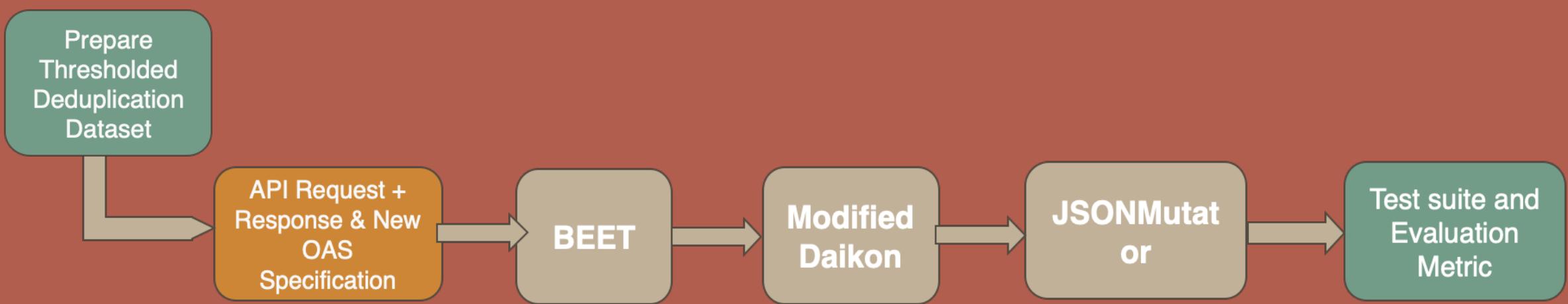
- AGORA should generate the new unique invariants after removing or reducing number of



Plan

Experiment 2

- In this experiment, we will perform exactly the same steps as Experiment 1 but will provide a different OAS specification that we are going to modify for another research question and record our observations.
- For removing duplicates, we will use the duplication removal utility which takes threshold value as input and then prepare the data based on that threshold value.



Evaluation Metric

1. Number of unique invariants identified

This metric will help assess the impact of removing duplicates on the diversity and richness of the invariants generated. A higher number of unique invariants could indicate more thorough test coverage.

2. Mutation Score

Apply mutation testing to the application and use the test suite to detect the mutants. The mutation score (percentage of mutants detected) can be a robust indicator of the quality of the test suite.

Initial Findings – Experiment 1

Authors' Output - 50 duplicate **createPlaylist** REST API - 43 invariants

invariants_50_modified ~

Possible Data Loss: Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

Save As...

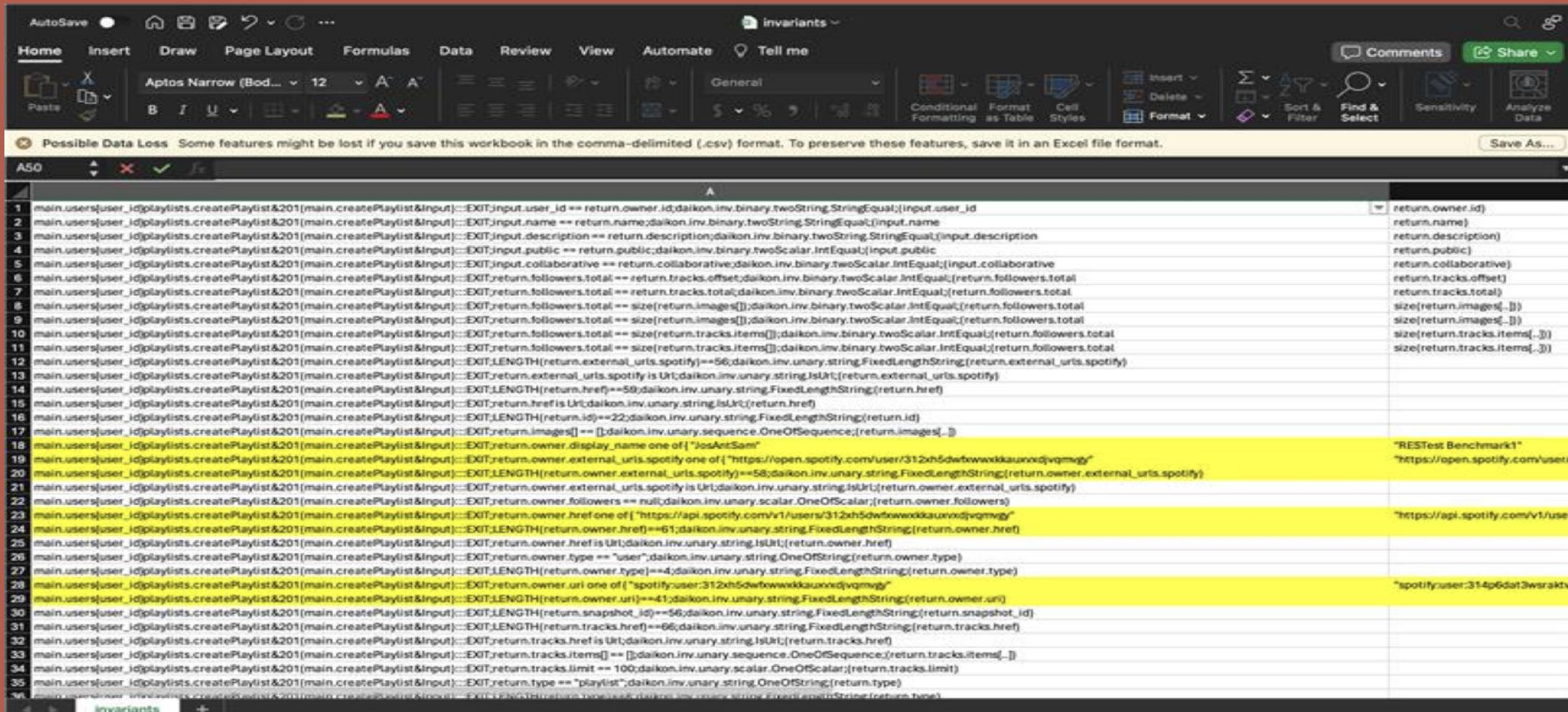
A	B
1 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;input.user_id == return.owner.id;daikon.inv.binary.twoString.StringEqual;(input.user_id	return.owner.id)
2 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;input.name == return.name;daikon.inv.binary.twoString.StringEqual;(input.name	return.name)
3 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;input.description == return.description;daikon.inv.binary.twoString.StringEqual;(input.description	return.description)
4 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;input.public == return.public;daikon.inv.binary.twoScalar.IntEqual;(input.public	return.public)
5 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;input.collaborative == return.collaborative;daikon.inv.binary.twoScalar.IntEqual;(input.collaborative	return.collaborative)
6 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.followers.total == return.tracks.offset;daikon.inv.binary.twoScalar.IntEqual;(return.followers.total	return.tracks.offset)
7 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.followers.total == return.tracks.total;daikon.inv.binary.twoScalar.IntEqual;(return.followers.total	return.tracks.total)
8 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.followers.total == size(return.images[]);daikon.inv.binary.twoScalar.IntEqual;(return.followers.total	size(return.images[...])
9 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.followers.total == size(return.images[]);daikon.inv.binary.twoScalar.IntEqual;(return.followers.total	size(return.images[...])
10 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.followers.total == size(return.tracks.items[]);daikon.inv.binary.twoScalar.IntEqual;(return.followers.total	size(return.tracks.items[...])
11 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.followers.total == size(return.tracks.items[]);daikon.inv.binary.twoScalar.IntEqual;(return.followers.total	size(return.tracks.items[...])
12 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;LENGTH(return.external_urls.spotify)==56;daikon.inv.unary.string.FixedLengthString;(return.external_urls.spotify)	
13 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.external_urls.spotify is Uri;daikon.inv.unary.string.isUri;(return.external_urls.spotify)	
14 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;LENGTH(return.href)==50;daikon.inv.unary.string.FixedLengthString;(return.href)	
15 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.href is Uri;daikon.inv.unary.string.isUri;(return.href)	
16 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;LENGTH(return.id)==22;daikon.inv.unary.string.FixedLengthString;(return.id)	
17 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.images[] == [];daikon.inv.unary.sequence.OneOfSequence;(return.images[...])	
18 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.owner.external_urls.spotify is Uri;daikon.inv.unary.string.isUri;(return.owner.external_urls.spotify)	
19 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.owner.followers == null;daikon.inv.unary.scalar.OneOfScalar;(return.owner.followers)	
20 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.owner.href is Uri;daikon.inv.unary.string.isUri;(return.owner.href)	
21 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.owner.type == "user";daikon.inv.unary.string.OneOfString;(return.owner.type)	
22 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;LENGTH(return.owner.type)==4;daikon.inv.unary.string.FixedLengthString;(return.owner.type)	
23 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;LENGTH(return.snapshot_id)==56;daikon.inv.unary.string.FixedLengthString;(return.snapshot_id)	
24 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;LENGTH(return.tracks.href)==66;daikon.inv.unary.string.FixedLengthString;(return.tracks.href)	
25 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.tracks.href is Uri;daikon.inv.unary.string.isUri;(return.tracks.href)	
26 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.tracks.items[] == [];daikon.inv.unary.sequence.OneOfSequence;(return.tracks.items[...])	
27 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.tracks.limit == 100;daikon.inv.unary.scalar.OneOfScalar;(return.tracks.limit)	
28 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;return.type == "playlist";daikon.inv.unary.string.OneOfString;(return.type)	
29 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;LENGTH(return.type)==8;daikon.inv.unary.string.FixedLengthString;(return.type)	
30 main.users[user_id]playlists.createPlaylist&201(main.createPlaylist&Input)::EXIT;LENGTH(return.uri)==39;daikon.inv.unary.string.FixedLengthString;(return.uri)	

Invariants_50_modified

Ready Accessibility: Unavailable 100%

Initial Findings – Experiment 1

Experiment 1 - 5 (threshold value) duplicate REST apis - 50 invariants
Findings - 7 new invariants found.

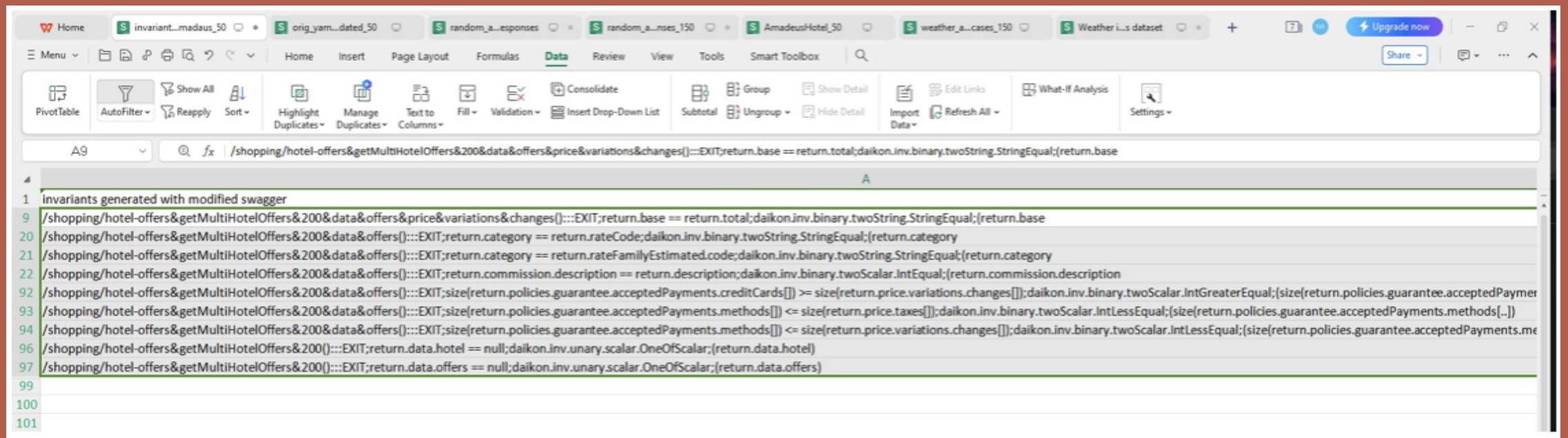


1	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <input.user_id =="return.owner.id>daikon.inv.binary.twoString.StringEqual();input.user_id</td"><td>return.owner.id)</td></input.user_id>	return.owner.id)
2	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <input.name =="return.name>daikon.inv.binary.twoString.StringEqual();input.name</td"><td>return.name)</td></input.name>	return.name)
3	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <input.description =="return.description>daikon.inv.binary.twoString.StringEqual();input.description</td"><td>return.description)</td></input.description>	return.description)
4	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <input.public =="return.public>daikon.inv.binary.twoScalar.IntEqual();input.public</td"><td>return.public)</td></input.public>	return.public)
5	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <input.collaborative =="return.collaborative>daikon.inv.binary.twoScalar.IntEqual();input.collaborative</td"><td>return.collaborative)</td></input.collaborative>	return.collaborative)
6	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.followers.total =="return.tracks.offset>daikon.inv.binary.twoScalar.IntEqual();return.followers.total</td"><td>return.tracks.offset)</td></return.followers.total>	return.tracks.offset)
7	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.followers.total =="return.tracks.total>daikon.inv.binary.twoScalar.IntEqual();return.followers.total</td"><td>return.tracks.total)</td></return.followers.total>	return.tracks.total)
8	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.followers.total =="size(return.images)>daikon.inv.binary.twoScalar.IntEqual();return.followers.total</td"><td>size(return.images))</td></return.followers.total>	size(return.images))
9	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.followers.total =="size(return.images)>daikon.inv.binary.twoScalar.IntEqual();return.followers.total</td"><td>size(return.images))</td></return.followers.total>	size(return.images))
10	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.followers.total =="size(return.tracks.items)>daikon.inv.binary.twoScalar.IntEqual();return.followers.total</td"><td>size(return.tracks.items))</td></return.followers.total>	size(return.tracks.items))
11	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.followers.total =="size(return.tracks.items)>daikon.inv.binary.twoScalar.IntEqual();return.followers.total</td"><td>size(return.tracks.items))</td></return.followers.total>	size(return.tracks.items))
12	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <length(<return.external_uris.spotify)==56>daikon.inv.unary.string.fixedlengthstring(<return.external_uris.spotify)< td=""><td>size(return.external_uris.spotify))</td></length(<return.external_uris.spotify)==56>daikon.inv.unary.string.fixedlengthstring(<return.external_uris.spotify)<>	size(return.external_uris.spotify))
13	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.external_uris.spotify =="Uri>daikon.inv.unary.string.IsUri(<return.external_uris.spotify)</td"><td>"https://open.spotify.com/user/</td></return.external_uris.spotify>	"https://open.spotify.com/user/
14	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <length(<return.href)==59>daikon.inv.unary.string.fixedlengthstring(<return.href)< td=""><td>"https://open.spotify.com/user/</td></length(<return.href)==59>daikon.inv.unary.string.fixedlengthstring(<return.href)<>	"https://open.spotify.com/user/
15	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.href =="Uri>daikon.inv.unary.string.IsUri(<return.href)</td"><td>"https://open.spotify.com/user/</td></return.href>	"https://open.spotify.com/user/
16	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <length(<return.id)==22>daikon.inv.unary.string.fixedlengthstring(<return.id)< td=""><td>"https://api.spotify.com/v1/users/</td></length(<return.id)==22>daikon.inv.unary.string.fixedlengthstring(<return.id)<>	"https://api.spotify.com/v1/users/
17	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.images>[] == []>daikon.inv.unary.sequence.OneOfSequence(<return.images))< td=""><td>"https://api.spotify.com/v1/users/</td></return.images))<></return.images>	"https://api.spotify.com/v1/users/
18	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.owner.display_name ["josantsam"]<="" oneof="" td=""><td>"RESTest Benchmark1"</td></return.owner.display_name>	"RESTest Benchmark1"
19	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.owner.external_uris.spotify 312xh5dwfwowkkauuoxdjqvqmgjy"}<="" oneof="" open.spotify.com="" td="" user="" {"https:=""><td>"https://open.spotify.com/user/</td></return.owner.external_uris.spotify>	"https://open.spotify.com/user/
20	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <length(<return.owner.external_uris.spotify)==58>daikon.inv.unary.string.fixedlengthstring(<return.owner.external_uris.spotify)< td=""><td>"https://open.spotify.com/user/</td></length(<return.owner.external_uris.spotify)==58>daikon.inv.unary.string.fixedlengthstring(<return.owner.external_uris.spotify)<>	"https://open.spotify.com/user/
21	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.owner.external_uris.spotify =="Uri>daikon.inv.unary.string.IsUri(<return.owner.external_uris.spotify)</td"><td>"https://open.spotify.com/user/</td></return.owner.external_uris.spotify>	"https://open.spotify.com/user/
22	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.owner.followers =="null>daikon.inv.unary.scalar.OneOfScalar(<return.owner.followers)</td"><td>"https://api.spotify.com/v1/users/</td></return.owner.followers>	"https://api.spotify.com/v1/users/
23	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.owner.href 312xh5dwfwowkkauuoxdjqvqmgjy"}<="" api.spotify.com="" oneof="" td="" users="" v1="" {"https:=""><td>"https://api.spotify.com/v1/users/</td></return.owner.href>	"https://api.spotify.com/v1/users/
24	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <length(<return.owner.href)==61>daikon.inv.unary.string.fixedlengthstring(<return.owner.href)< td=""><td>"https://api.spotify.com/v1/users/</td></length(<return.owner.href)==61>daikon.inv.unary.string.fixedlengthstring(<return.owner.href)<>	"https://api.spotify.com/v1/users/
25	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.owner.href =="Uri>daikon.inv.unary.string.IsUri(<return.owner.href)</td"><td>"https://api.spotify.com/v1/users/</td></return.owner.href>	"https://api.spotify.com/v1/users/
26	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.owner.type >daikon.inv.unary.string.oneofstring(<return.owner.type)<="" =="user" td=""><td>"https://api.spotify.com/v1/users/</td></return.owner.type>	"https://api.spotify.com/v1/users/
27	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <length(<return.owner.type)==4>daikon.inv.unary.string.fixedlengthstring(<return.owner.type)< td=""><td>"https://api.spotify.com/v1/users/</td></length(<return.owner.type)==4>daikon.inv.unary.string.fixedlengthstring(<return.owner.type)<>	"https://api.spotify.com/v1/users/
28	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.owner.url oneof="" td="" {"spotify:user:312xh5dwfwowkkauuoxdjqvqmgjy"}<=""><td>"https://api.spotify.com/v1/users/</td></return.owner.url>	"https://api.spotify.com/v1/users/
29	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <length(<return.owner.url)==41>daikon.inv.unary.string.fixedlengthstring(<return.owner.url)< td=""><td>"https://api.spotify.com/v1/users/</td></length(<return.owner.url)==41>daikon.inv.unary.string.fixedlengthstring(<return.owner.url)<>	"https://api.spotify.com/v1/users/
30	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <length(<return.snapshot_id)==56>daikon.inv.unary.string.fixedlengthstring(<return.snapshot_id)< td=""><td>"https://api.spotify.com/v1/users/</td></length(<return.snapshot_id)==56>daikon.inv.unary.string.fixedlengthstring(<return.snapshot_id)<>	"https://api.spotify.com/v1/users/
31	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <length(<return.tracks.href)==66>daikon.inv.unary.string.fixedlengthstring(<return.tracks.href)< td=""><td>"https://api.spotify.com/v1/users/</td></length(<return.tracks.href)==66>daikon.inv.unary.string.fixedlengthstring(<return.tracks.href)<>	"https://api.spotify.com/v1/users/
32	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.tracks.href =="Uri>daikon.inv.unary.string.IsUri(<return.tracks.href)</td"><td>"https://api.spotify.com/v1/users/</td></return.tracks.href>	"https://api.spotify.com/v1/users/
33	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.tracks.items>[] == []>daikon.inv.unary.sequence.OneOfSequence(<return.tracks.items))< td=""><td>"https://api.spotify.com/v1/users/</td></return.tracks.items))<></return.tracks.items>	"https://api.spotify.com/v1/users/
34	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.tracks.limit =="100>daikon.inv.unary.scalar.OneOfScalar(<return.tracks.limit)</td"><td>"https://api.spotify.com/v1/users/</td></return.tracks.limit>	"https://api.spotify.com/v1/users/
35	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <return.type >daikon.inv.unary.string.oneofstring(<return.type)<="" =="playlist" td=""><td>"https://api.spotify.com/v1/users/</td></return.type>	"https://api.spotify.com/v1/users/
36	main.users\$user_id\$playlists.createPlaylistÉ(main.createPlaylist&Input)::=EXIT <length(<return.tracks_type)==10>daikon.inv.unary.string.fixedlengthstring(<return.tracks_type))< td=""><td>"https://api.spotify.com/v1/users/</td></length(<return.tracks_type)==10>daikon.inv.unary.string.fixedlengthstring(<return.tracks_type))<>	"https://api.spotify.com/v1/users/

RQ - 3 : Investigating the Impact of Dataset and Time-based Mutations on AGORA's Fault Detection

Purpose: To explore the impact of various factors on invariant generation and AGORA's failure detection capabilities.

Approach: Conducted three experiments to analyze the effects of dataset changes and time-based mutations.



The screenshot shows a Microsoft Excel spreadsheet with the Data tab selected. The formula bar displays a complex Java-like code snippet involving daikon.inv.binary.twoString.StringEqual and return.base. The main content area contains a list of numbered invariants, each starting with a line number and a brief description. The invariants are generated from modified swagger files, specifically for shopping/hotel-offers endpoints. The list includes various assertions about the structure and values of returned objects like total, category, rateCode, rateFamilyEstimated.code, commission.description, policies.guarantee.acceptedPayments.creditCards.size, price.variations.changes.size, and data.hotel/offer sizes.

Line Number	Invariant Description
1	invariants generated with modified swagger
9	/shopping/hotel-offers&getMultiHotelOffers&200&data&offers&price&variations&changes():::EXIT;return.base == return.total;daikon.inv.binary.twoString.StringEqual;(return.base
20	/shopping/hotel-offers&getMultiHotelOffers&200&data&offers():::EXIT;return.category == return.rateCode;daikon.inv.binary.twoString.StringEqual;(return.category
21	/shopping/hotel-offers&getMultiHotelOffers&200&data&offers():::EXIT;return.category == return.rateFamilyEstimated.code;daikon.inv.binary.twoString.StringEqual;(return.category
22	/shopping/hotel-offers&getMultiHotelOffers&200&data&offers():::EXIT;return.commission.description == return.description;daikon.inv.binary.twoScalar.IntEqual;(return.commission.description
92	/shopping/hotel-offers&getMultiHotelOffers&200&data&offers():::EXIT;size{return.policies.guarantee.acceptedPayments.creditCards[]} >= size{return.price.variations.changes[]};daikon.inv.binary.twoScalar.IntGreaterEqual;(size{return.policies.guarantee.acceptedPayments.creditCards[]})
93	/shopping/hotel-offers&getMultiHotelOffers&200&data&offers():::EXIT;size{return.policies.guarantee.acceptedPayments.methods[]} <= size{return.price.taxes[]};daikon.inv.binary.twoScalar.IntLessEqual;(size{return.policies.guarantee.acceptedPayments.methods[]})
94	/shopping/hotel-offers&getMultiHotelOffers&200&data&offers():::EXIT;size{return.policies.guarantee.acceptedPayments.methods[]} <= size{return.price.variations.changes[]};daikon.inv.binary.twoScalar.IntLessEqual;(size{return.policies.guarantee.acceptedPayments.methods[]})
96	/shopping/hotel-offers&getMultiHotelOffers&200():::EXIT;return.data.hotel == null;daikon.inv.unary.scalar.OneOfScalar;(return.data.hotel)
97	/shopping/hotel-offers&getMultiHotelOffers&200():::EXIT;return.data.offers == null;daikon.inv.unary.scalar.OneOfScalar;(return.data.offers)
99	
100	
101	

Snapshot of the findings from experiment 1

Experiment 1: Updating OpenAPI Format with Existing Dataset

Details:

Utilized the Amadeus Hotel API for comparison between different versions of Swagger files.

Highlighted Changes:

- Version Update: Transition from 3.0.5 to 3.0.8 with potential bug fixes or updates.
- Consistent Endpoints: Stability observed in API functional range.
- Parameter Adjustments: Minor modifications noted impacting invariant generation.
- Enhanced Enumerations and Additional Properties: Improvements in data categorization and description.

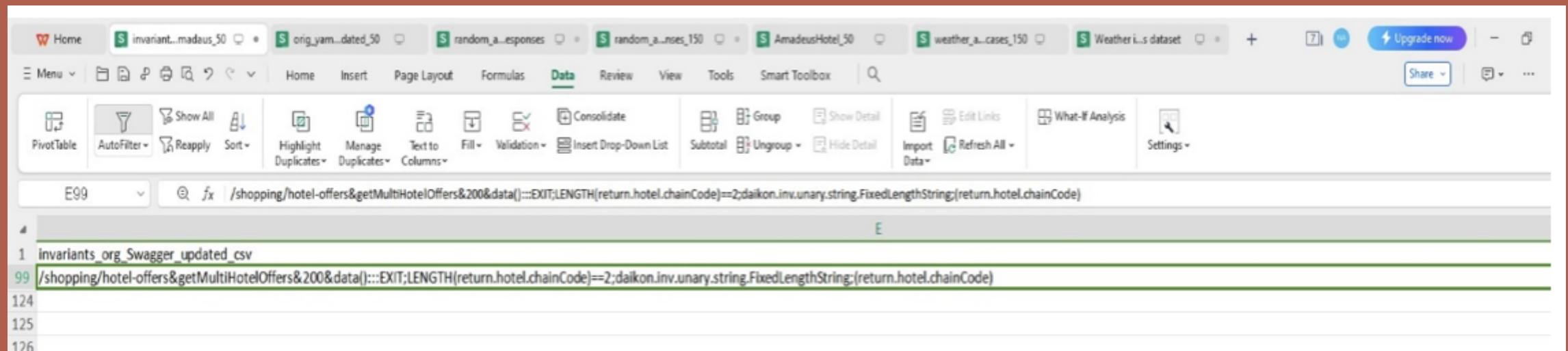
Analysis Outcome:

Daikon identified fewer invariants in the modified version, mainly due to parameter changes.

Experiment 2 : Updating API Responses with Previous Version of OpenAPI Format.

Details:

- Kept OpenAPI version constant while altering API response parameters for the Amadeus Hotel API.
- Minimal differences observed in generated invariants between original and modified versions.
- Conclusion: Minor modifications in API responses have limited impact on invariant generation and fault detection ratio.



The screenshot shows a Microsoft Excel spreadsheet with the following details:

- Excel ribbon:** The ribbon is visible at the top, showing tabs for Home, Insert, Page Layout, Formulas, Data, Review, View, Tools, Smart Toolbox, and a search bar.
- Data tab:** The Data tab is selected, showing various data analysis tools like PivotTable, AutoFilter, and Consolidate.
- Cell E99:** The formula bar shows the formula: `/shopping/hotel-offers&getMultiHotelOffers&200&data()::EXIT;LENGTH(return.hotel.chainCode)==2;daikon.inv.unary.string.FixedLengthString;(return.hotel.chainCode)`.
- Content:** The spreadsheet contains two rows of text:
 - Row 1: `1 invariants_org_Swagger_updated_csv`
 - Row 99: `/shopping/hotel-offers&getMultiHotelOffers&200&data()::EXIT;LENGTH(return.hotel.chainCode)==2;daikon.inv.unary.string.FixedLengthString;(return.hotel.chainCode)`
- Row Numbers:** Row numbers 124, 125, and 126 are visible on the left side.

Experiment 3: Time Based Mutation Analysis for HRMS API

Details:

1. Importance of Time-based Mutations for APIs with time-sensitive data.

2. Examples of Mutations:

- Adjusting Timestamps
- Modifying Durations and Intervals
- Changing Expiration Dates

3. Implementation Steps:

- Identifying Time-Sensitive Data
- Developing Custom Mutation Operators
- Mutation Testing with AGORA

4. Expected Outcome: AGORA should identify faults assuming accurate time-based invariants generated by Daikon.

**THANK
YOU**