



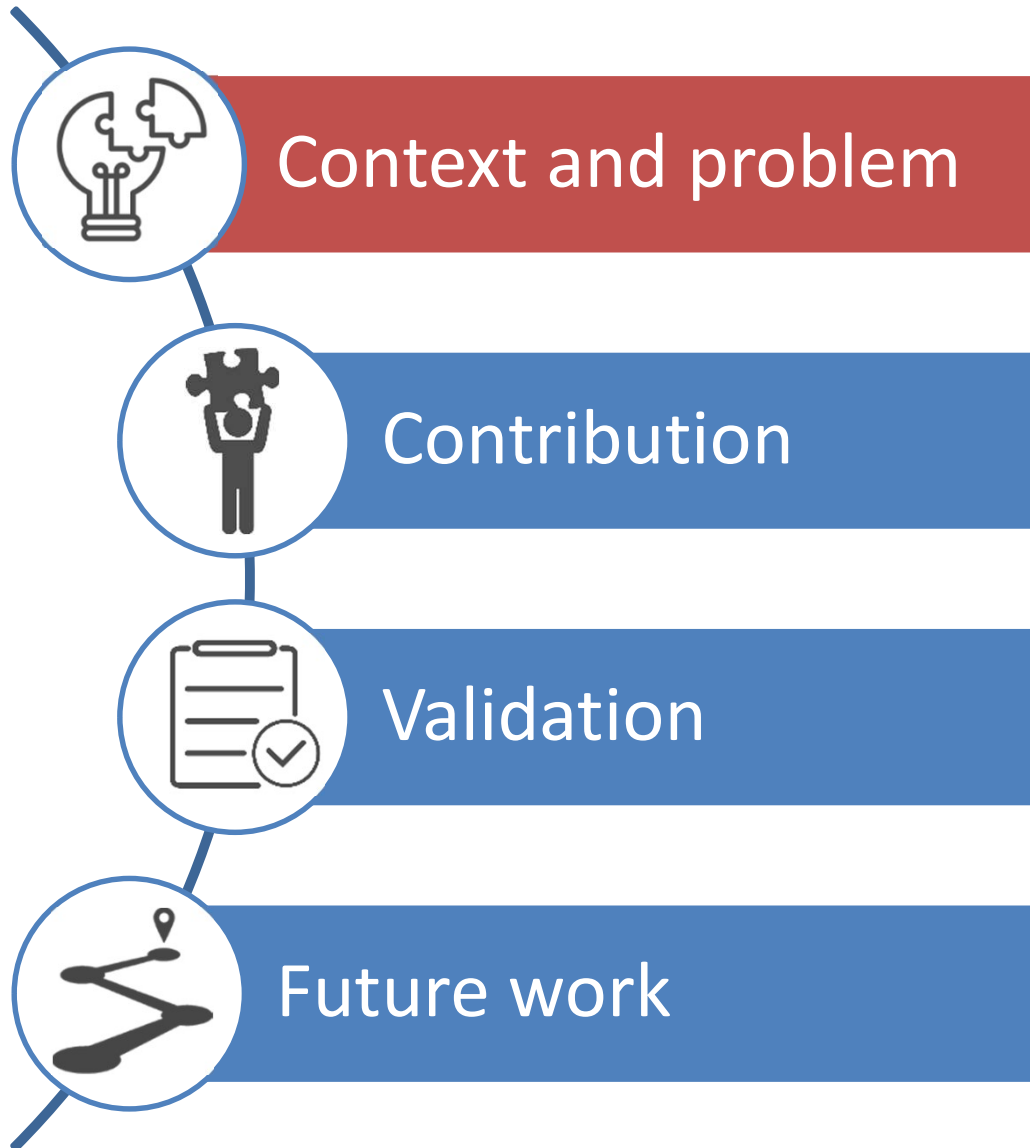
Automated Generation of Realistic Test Inputs for Web APIs

Juan Carlos Alonso Valenzuela

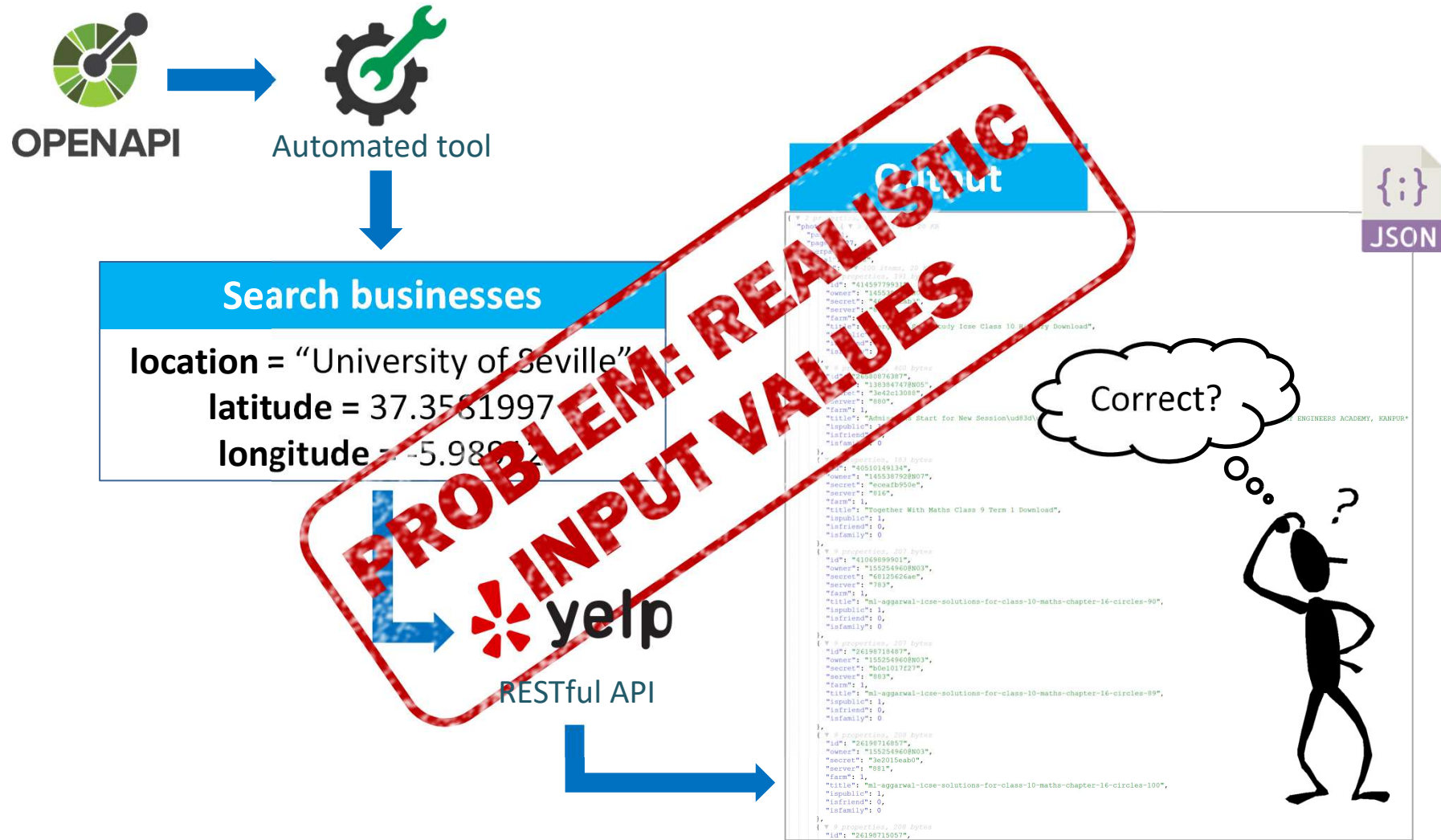
[UNDERGRADUATE]

SCORE Lab, University of Seville, Spain

ESEC/FSE SRC
August 2021



Black-box testing of Web APIs



Problem

Testing web APIs requires realistic Input values



Parameters

Name	Description
countryCode * required string (query)	2-letter ISO 3166-1 alpha-2 code specifying the country.
addressLocality string (query)	Text specifying the name of the locality, for example a city.
postalCode string (query)	Text specifying the postal code for an address.
streetAddress string (query)	The street address is expressed as free form text.



Name	Description
location	Geographic area.
latitude	Latitude of the location.
longitude	Longitude of the location.

amadeus

GET

/shopping/hotel-offers FIND HOTELS.

Name	Description
cityCode	Destination City Code (or Airport Code). In case of city code, the search will be done around the city center.
hotelName	Search by Hotel Name. Accepts maximum 4 keywords.
currency	Request a specific currency. ISO currency code
lang	ISO language code

Related work

Querying knowledge bases for test input generation (GUI)



Link: Exploiting the Web of Data to Generate Test Inputs

Leonardo Mariani[§] Mauro Pezzè^{†§} Oliviero Riganelli[§] Mauro Santoro[§]

[§] Department of Informatics, Systems and Communications
University of Milano Bicocca - Milano, Italy
{mariani,pezze,riganelli,santoro}@disco.unimib.it

[†] Faculty of Informatics
University of Lugano - Lugano, Switzerland
mauro.pezze@usi.ch

- **LINK:** Query DBpedia to generate input values



- **SAIGEN:** Extension of Link for mobile apps



Testing Apps With Real World Inputs

Tanapuch Wanwarang
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany
tanapuch.wanwarang@cispa.saarland

Leon Bettscheider
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany
s8lnbett@stud.uni-saarland.de

Nataniel P. Borges Jr.
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany
nataniel.borges@cispa.saarland

Andreas Zeller
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany
zeller@cispa.saarland

Related Work

Unique challenges of Web APIs



No GUI



OMDb API

```
parameters:  
- name: t  
description: 'Introduce a movie title.'  
in: query  
required: true  
type: string
```

Unspecific parameter names



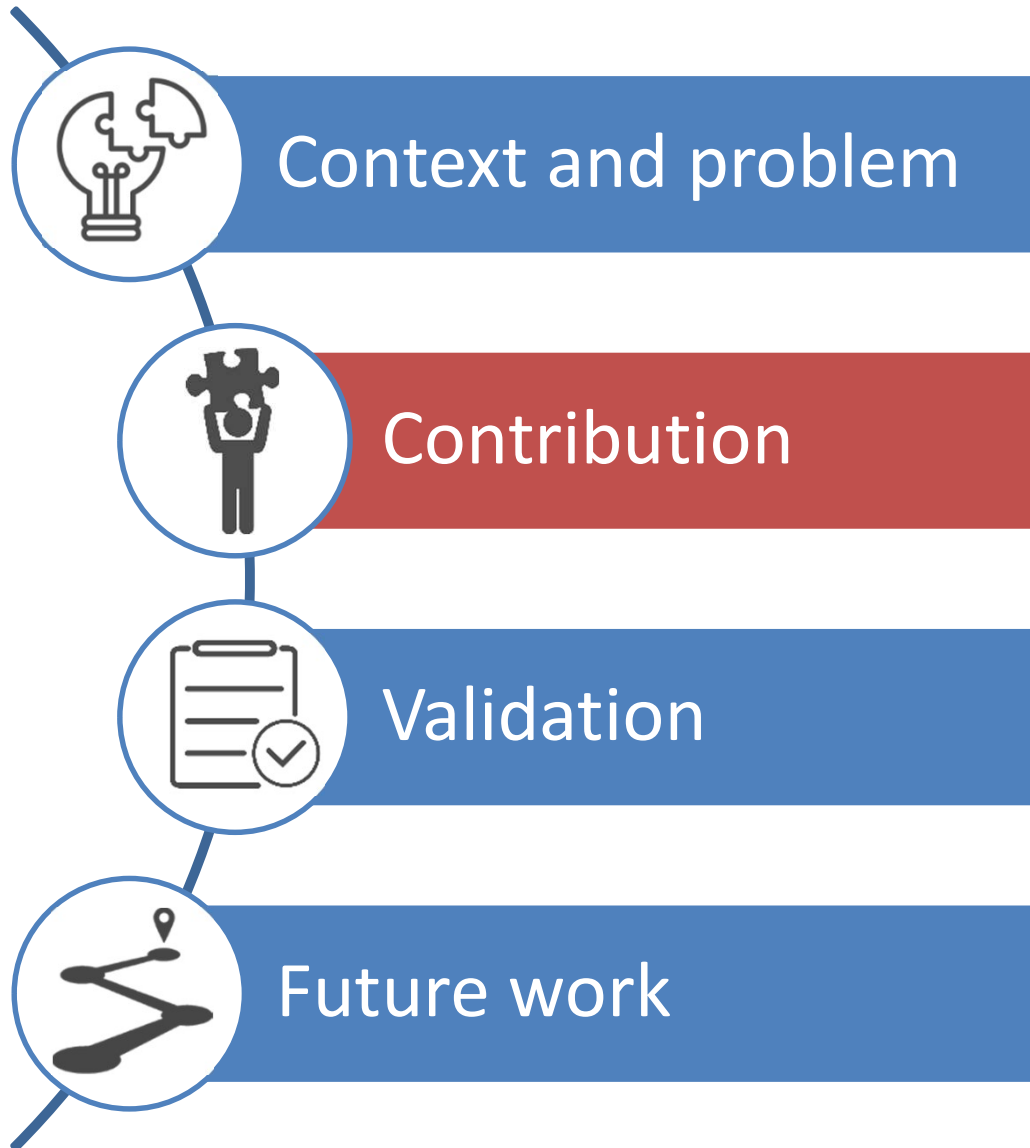
amadeus

```
parameters:  
- name: currency  
description: 'Introduce a valid currency code.'  
in: query  
required: true  
type: string
```

Descriptions in
natural language

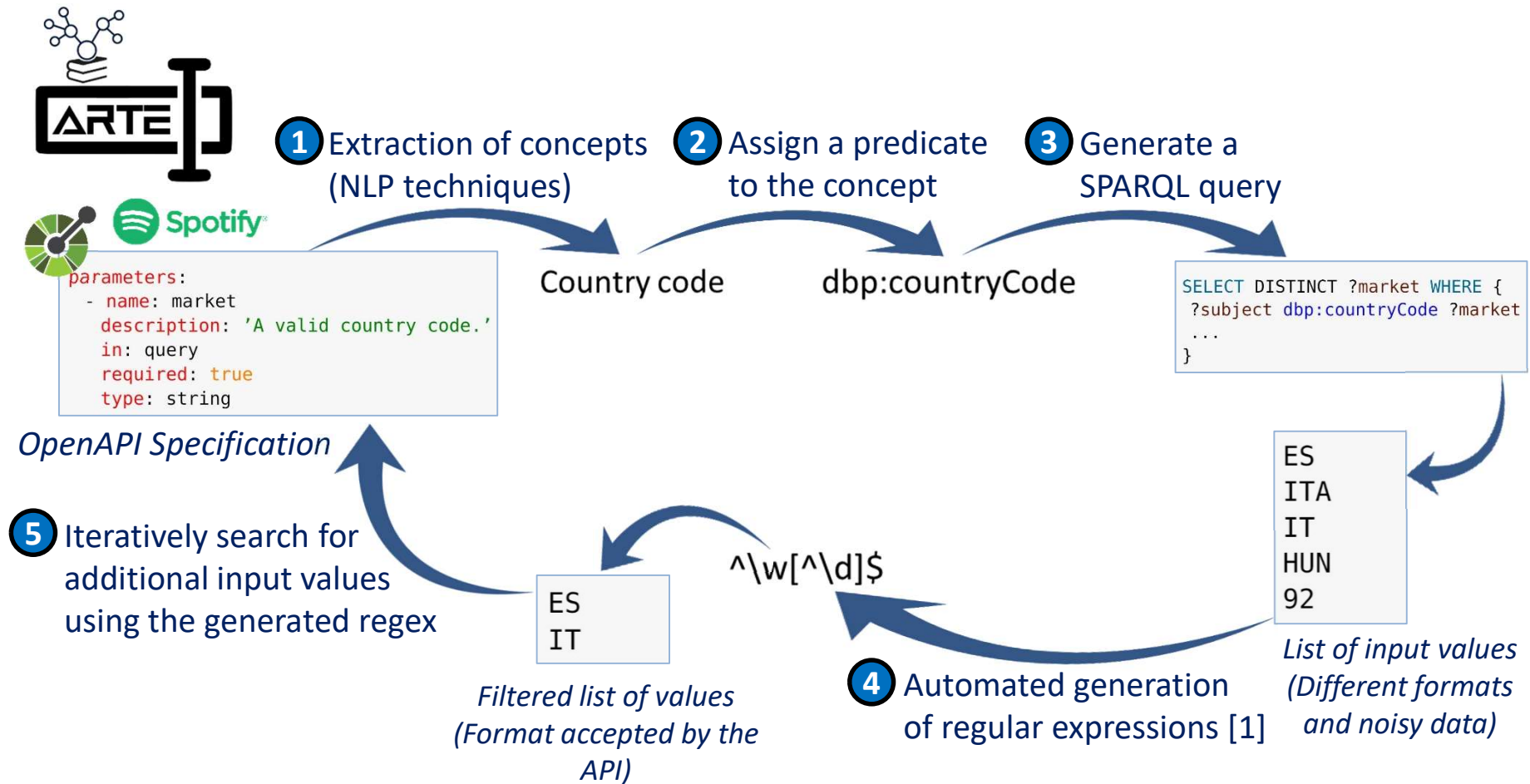
```
- name: postal_code  
- name: postalCode  
- name: postal-code
```

Different naming
conventions



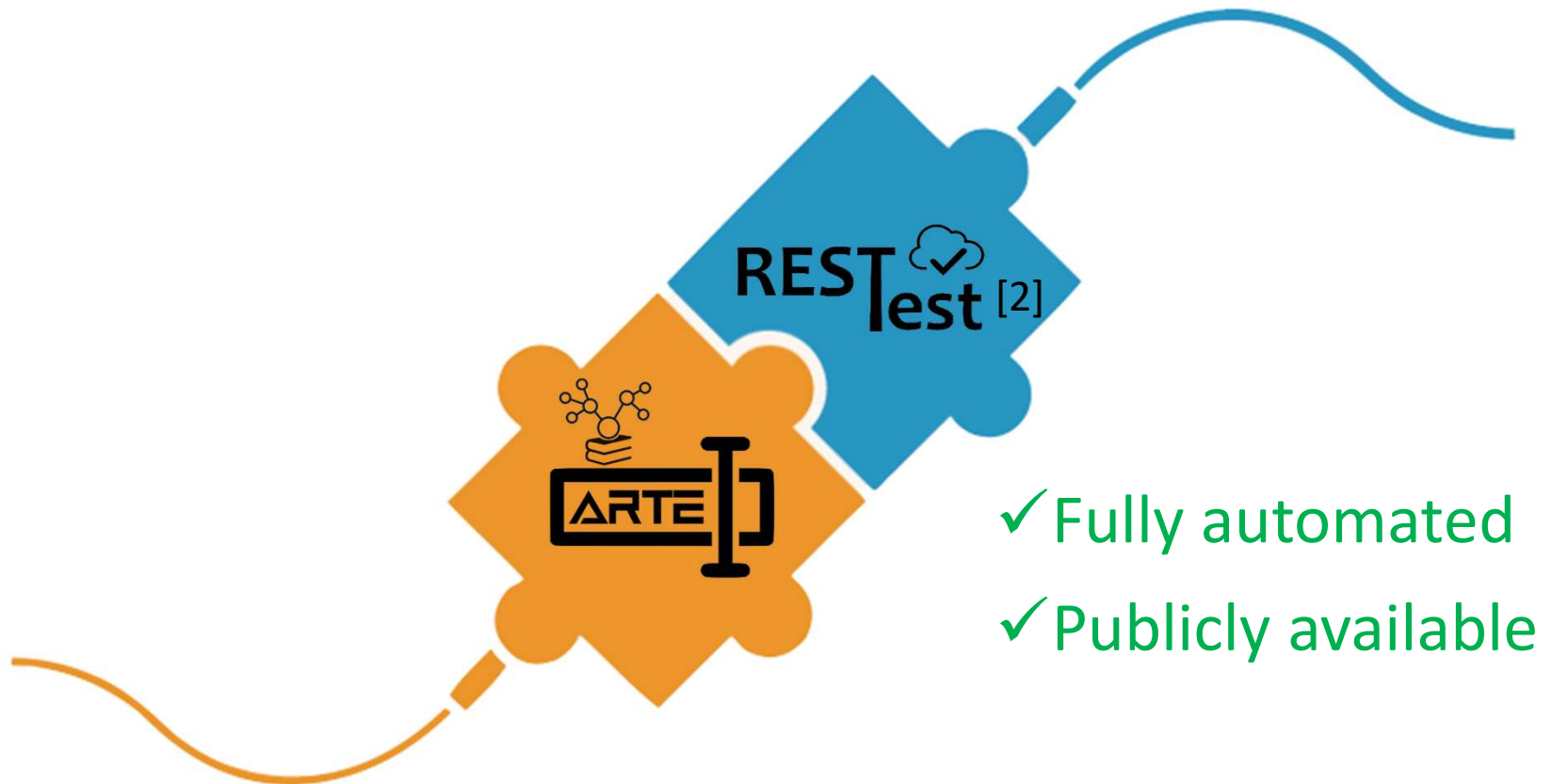
Contribution

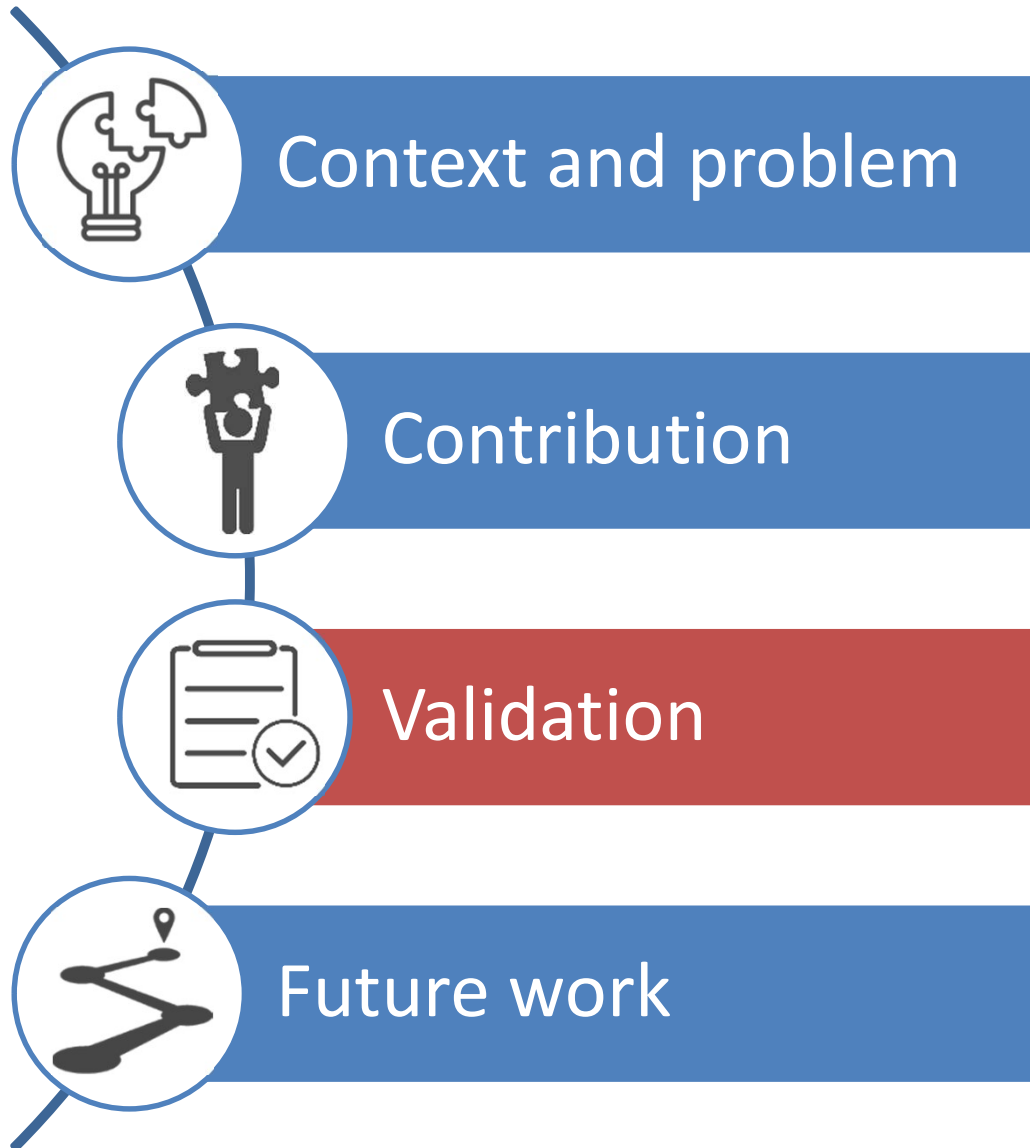
ARTE: Automated generation of Realistic Test inputs



Contribution

Integrated into state-of-the-art framework





amadeus



OMDb API



Validation

Experiment 1: Generation of Realistic Input values

26 APIs – **83** Operations – **99** Parameters

38.9%
SAIGEN

66.7%
ARTE

amadeus



OMDb API



Validation

Experiment 1: Generation of Realistic Input values

26 APIs – **83** Operations – **99** Parameters

38.9%
SAIGEN

66.7%
ARTE

47 APIs – **136** Operations – **207** Parameters

New Results

30.9%
SAIGEN

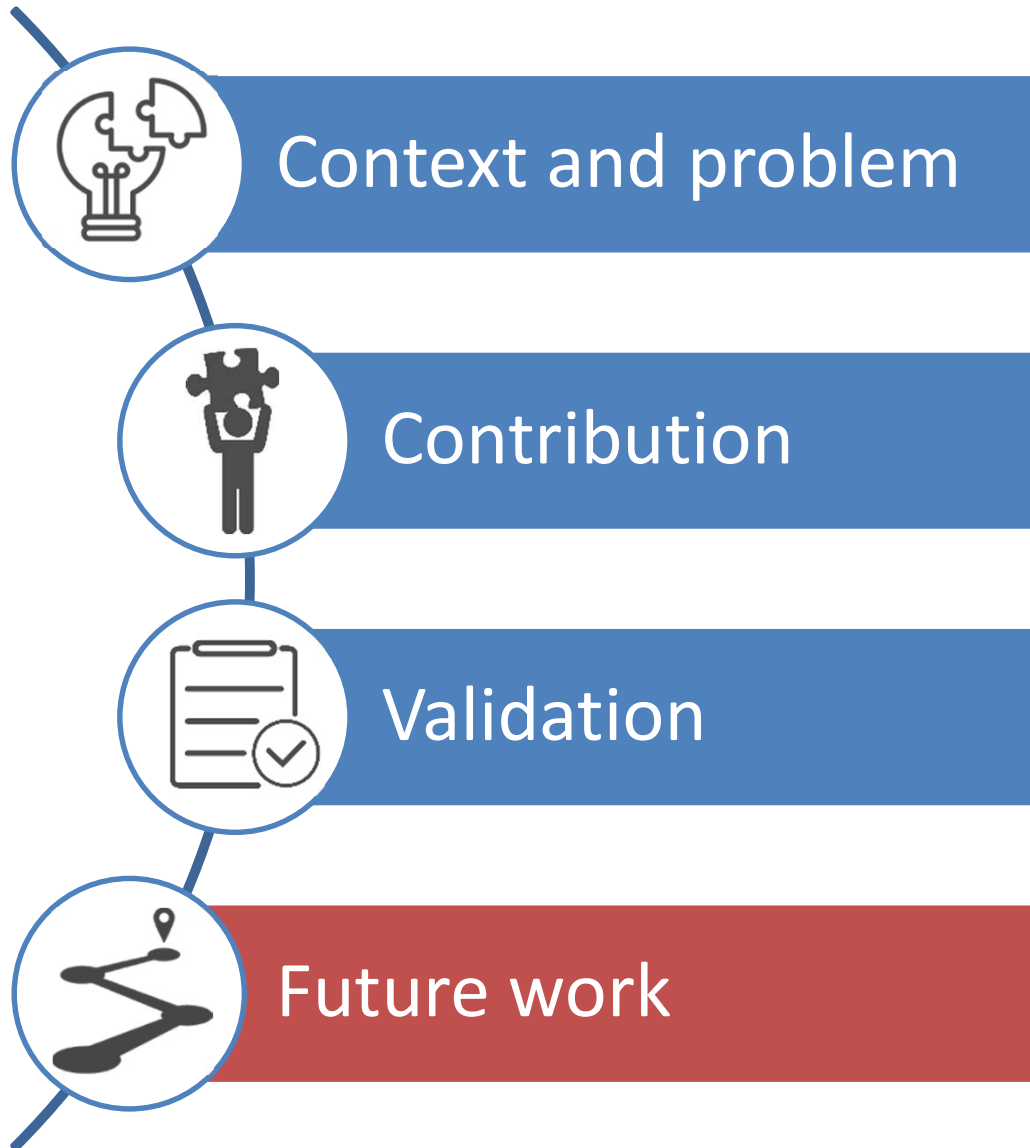
64.3%
ARTE

Validation

Experiment 2: Generation of Valid API Requests

6 Industrial APIs – **13** Operations – **39** Parameters

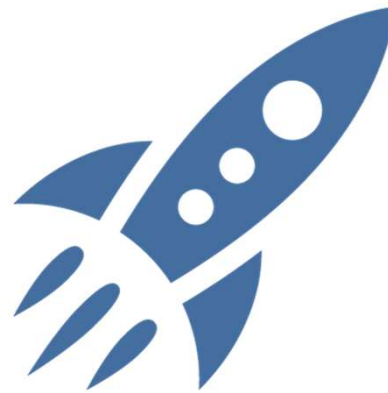




Future Work



Learn from
previous API calls



Deployment as a
publicly available API



Generation of
Test Oracles

Thanks!



www.isa.us.es



Replication package



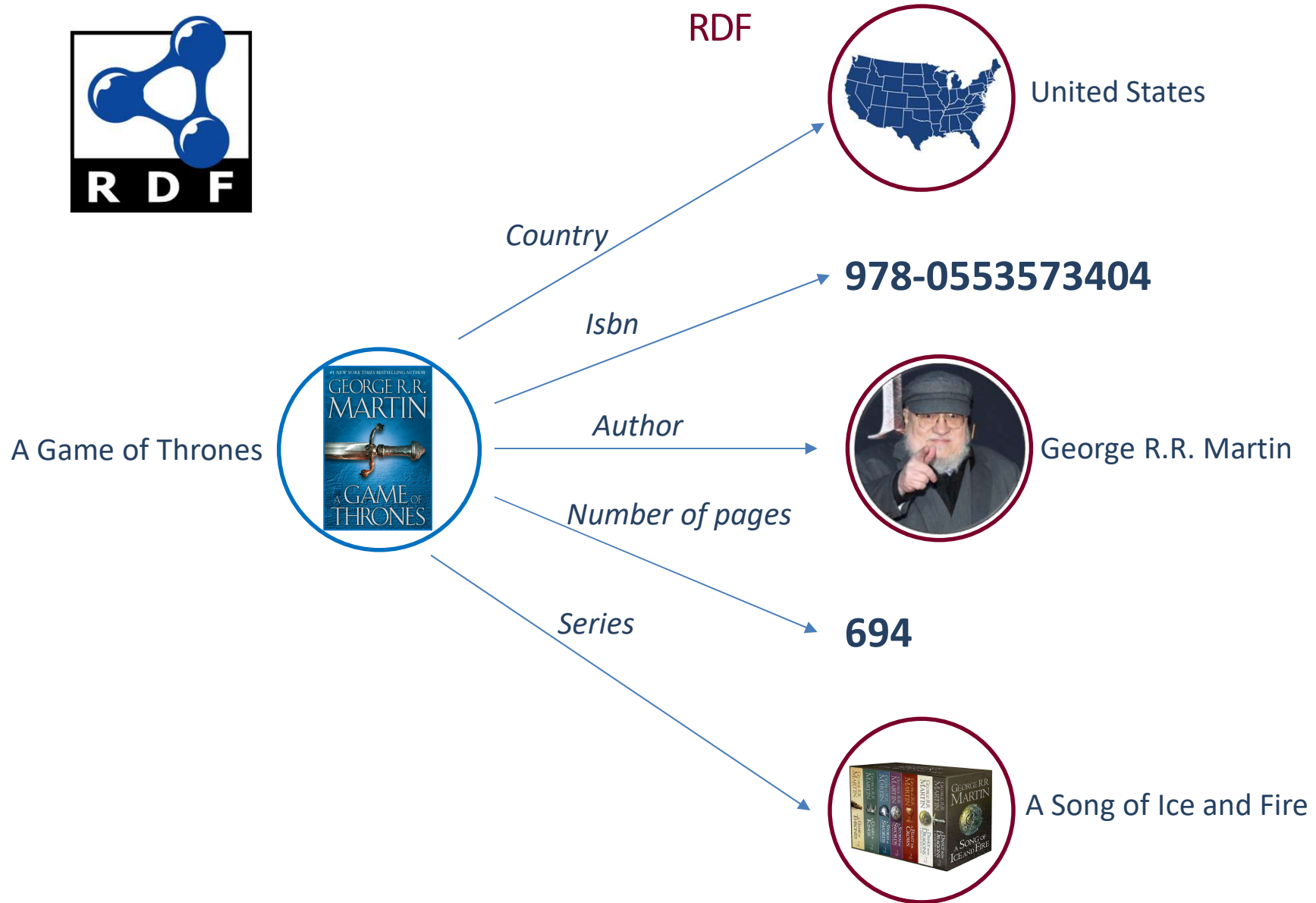
javalenzuela@us.es

Backup slides



Web of data

RDF



Web of data

URIs



A Game of Thrones

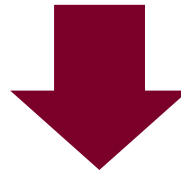


http://dbpedia.org/resource/A_Game_of_Thrones



Subject

Predicate



Author

<http://dbpedia.org/ontology/author>

George R.R. Martin



http://dbpedia.org/resource/George_R._R._Martin



Object

Web of data

SPARQL

SPARQL query

```
1 SELECT DISTINCT ?title ?author ?isbn WHERE {  
2   ?subject <http://dbpedia.org/property/title> ?title ;  
3     <http://dbpedia.org/ontology/author> ?author ;  
4     <http://dbpedia.org/ontology/isbn> ?isbn .  
5  
6   FILTER regex(str(?isbn), '^[0-9]*[-| ]{4}[0-9]*$')  
7 }
```



Results

Title	Author	ISBN
The Last Wish	http://dbpedia.org/resource/Andrzej Sapkowski	978-0-575-08244-1
The Martian	http://dbpedia.org/resource/Andy Weir	978-0-8041-3902-1
The Name of the Wind	http://dbpedia.org/resource/Patrick Rothfuss	978-0-7564-0407-9
An Oracle	http://dbpedia.org/resource/Sakyo Komatsu	978-4-7700-2039-0
The Black Tulip	http://dbpedia.org/resource/Alexandre Dumas	978-0-14-044892-4

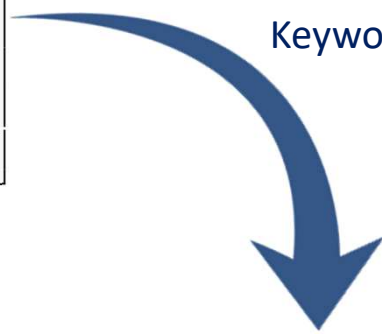
Predicate search

(Exhaustive) Search for predicates

SPARQL query used in the search for predicates

```
1 SELECT DISTINCT ?predicate WHERE {  
2   ?predicate a rdf:Property  
3  
4   FILTER regex(str(?predicate), keyword, 'i')  
5 } ORDER BY strlen(str(?predicate))
```

Keyword = currency



Results

Predicates
http://dbpedia.org/property/currency
http://dbpedia.org/ontology/currency
http://dbpedia.org/property/currencyIso
http://dbpedia.org/ontology/currencyCode
http://dbpedia.org/property/msrpCurrency

Predicate search

(Exhaustive) Search for predicates

SPARQL query used in the search for predicates

```
1 SELECT DISTINCT ?predicate WHERE {  
2   ?predicate a rdf:Property  
3  
4   FILTER regex(str(?predicate), keyword, 'i')  
5 } ORDER BY strlen(str(?predicate))
```

Keyword = airline code



Results

Predicates

<http://dbpedia.org/ontology/iataAirlineCode>

<http://dbpedia.org/ontology/icaoAirlineCode>

NLP techniques

Preprocessing

This is a sentences, showing off the stop words filtration

TOKENIZATION



[this, is, a, sentences, showing, off, the, stop, words, filtration]

POS TAGGING



[(this, DT), (is, VBZ), (a, DT), (sentences, NNS), (showing, VBG), (off, RP), (the, DT),
(stop, NN), (words, NNS), (filtration, NN)]

LEMMATIZATION



[this, is, a, sentence, show, off, the, stop, word, filtration]

STOP WORDS



[sentence, show, stop, word, filtration]

Processing parameter names

Parameter names with a single character



OMDb API

```
parameters:  
- name: t  
  description: 'Introduce a movie title.'  
  in: query  
  required: true  
  type: string
```



[introduce, movie, **title**]

Predicate search

Matching rules for obtaining keywords

1º Name of the parameter + code/id



```
parameters:  
- name: currency  
  description: 'Introduce a valid currency code.'  
  in: query  
  required: true  
  type: string
```



CurrencyCode

2º Word with the same initial characters as the parameter name + code/id



```
parameters:  
- name: lang  
  description: 'ISO language code.'  
  in: query  
  required: false  
  type: string
```

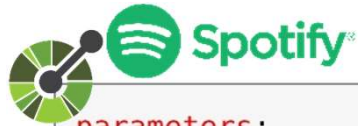


**LangCode
LanguageCode**

Predicate search

Matching rules for obtaining keywords

3º Noun/Foreign word + code/id



```
parameters:  
- name: market  
description: 'A valid country code.'  
in: query  
required: true  
type: string
```

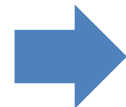


CountryCode

4º Unmodified parameter name



```
parameters:  
- name: location  
description: 'Geographic area.'  
in: query  
required: false  
type: string
```



Location

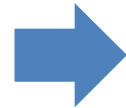
Predicate search

Matching rules for obtaining keywords

5º Convert naming conventions to camelCase

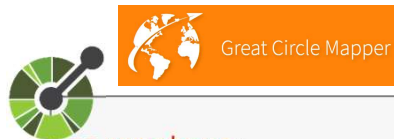


```
parameters:  
- name: zip_code  
  description: 'The zip of the business.'  
  in: query  
  required: true  
  type: string
```

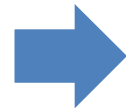


Zipcode

6º Split the parameter name into multiple words



```
parameters:  
- name: icao_iata  
  description: 'Insert icao or iata.'  
  in: query  
  required: true  
  type: string
```



**Icao
Iata**

Criteria used for creating the dataset

Discard reasons:

- APIs containing exclusively domain specific parameters and trivial parameters (such as enums)
- APIs without any validation
- APIs containing exclusively confidential parameters
- APIs that returned no response
- APIs without parameters
- Premium APIs
- APIs whose specification is not in english

Discard Reason	Number of discarded APIs
Only domain specific and trivial parameters	34
No validation	19
Confidential parameters	5
No response from the API	5
No parameters	5
Paid API	2
Unknown	1
Not in english	1
Total	72

Syntactically and Semantically valid values

Parameter name	Parameter value	Status code	Syntactically valid	Semantically valid
Address	742 Evergreen Terrace	200	✓	✓
Address	Dog	200	✓	✗
Currency code	USD	400	✗	✓
Currency code	Dog	400	✗	✗

Evaluation results

API	RapidAPI - ARTE				
	Operations	Parameters	Syntactically valid (%)	Semantically valid (%)	Syntactically and Semantically valid (%)
Airport info	1	2	100	100	100
API Basketball	5	6	100 (83.3)	33.3 (16,7)	33.3 (16.7)
API Football	5	4	100 (75)	75 (50)	75 (50)
Asos	5	6	66.7	50	50
Carbon Footprint	1	3	33.3	33.3	33.3
ClimaCell	3	2	100	100	100
Coronavirus map	4	1	0	100	0
Countries Cities	4	4	100	100	100
Flight data	9	6	66.7 (50)	66.7 (50)	66.7 (50)
Great Circle Mapper	3	3	33.3	33.3	33,3
Movie Database	2	2	100	50	50
Open Weather Map	4	3	100	100	100
Public Holiday	1	2	100	100	100
Similar Web	2	1	100	100	100
Skyscanner flight search	8	5	80 (40)	40 (0)	40 (0)
True Way Geocoding	2	4	75	50	50
Us Restaurant Menus	6	5	100	80	80
Us Weather by zipcode	1	1	100	100	100
Weather Forecast 14 days	3	5	100	80	80
Total	69	65	83 (75.4)	66.2 (58.5)	64.6 (56.9)

Evaluation results

API	RapidAPI - SAIGEN				
	Operations	Parameters	Syntactically valid (%)	Semantically valid (%)	Syntactically and Semantically valid (%)
Airport info	1	2	50	50	50
API Basketball	5	6	83.3	33.3	33.3
API Football	5	4	50	50	50
Asos	5	6	66.7	33.3	33.3
Carbon Footprint	1	3	33.3	33.3	33.3
ClimaCell	3	2	100	100	100
Coronavirus map	4	1	0	100	0
Countries Cities	4	4	75	75	75
Flight data	9	6	0	0	0
Great Circle Mapper	3	3	0	0	0
Movie Database	2	2	50	0	0
Open Weather Map	4	3	100	100	100
Public Holiday	1	2	50	50	50
Similar Web	2	1	0	0	0
Skyscanner flight search	8	5	40	0	0
True Way Geocoding	2	4	75	25	25
Us Restaurant Menus	6	5	80	60	60
Us Weather by zipcode	1	1	100	100	100
Weather Forecast 14 days	3	5	100	80	80
Total	69	65	58.5	41.5	40

Evaluation results

Industrial APIs - ARTE					
API	Operations	Parameters	Syntactically valid (%)	Semantically valid (%)	Syntactically and Semantically valid (%)
Amadeus Hotel	2	7	85.7	85.7	85.7
Deutschebahn StaDa	1	4	25	25	25
DHL Location Finder	2	6	100	100	100
Marvel	1	5	100	40	40
OMDb	1	3	100	100	100
Spotify	5	4	75	75	50
Yelp Fusion	2	5	80	80	80
Total	14	34	82.4	73.5	70,6

Evaluation results

Industrial APIs - SAIGEN					
API	Operations	Parameters	Syntactically valid (%)	Semantically valid (%)	Syntactically and Semantically valid (%)
Amadeus Hotel	2	7	42.9	42.9	42.9
Deutschebahn StaDa	1	4	0	0	0
DHL Location Finder	2	6	50	50	50
Marvel	1	5	60	40	40
OMDb	1	3	33.3	0	0
Spotify	5	4	25	25	0
Yelp Fusion	2	5	80	80	80
Total	14	34	44.1	38.2	36.3

Evaluation results

Experiment 2

Experiment 2: Generation of valid API calls				
API	Operation	Random (%)	SAIGEN (%)	ARTE (%)
Amadeus Hotel	Find Hotels	13	9.2	16.7
Amadeus Hotel	View hotel romos	44.5	23.4	60
Deutschebahn StaDa	Get stations	19	27.2	44.2
DHL Location Finder	Find by address	0	0.1	70
DHL Location Finder	Find by coordinates	0	97.9	100
OMDb	Search	40.1	34.9	35.1
Spotify	Get albums	47.9	49	95.4
Spotify	Get album	53.6	48.7	97.4
Spotify	Get categories	50	49.7	70.2
Spotify	Get category	48.7	51.4	74.5
Spotify	Get featured playlists	25.2	25.6	35.1
Yelp Fusion	Search businesses	31.4	50.9	48.3
Yelp Fusion	Search transactions	52.9	70.6	86
Total		32.8	41.1	64.1

Bugs detected

Amadeus Hotel

hotelName	Search by Hotel Name. Accepts maximum 4 keywords.
string	
(query)	Example: Hotel California Example: Hotel Califo

REQUEST

hotelName = "Waterfront Cebu City Hotel Casino"



RESPONSE

HTTP/1.1 400 Bad Request

Status code 400

Body

```
{
  "errors": [
    {
      "status": 400,
      "code": 1359,
      "title": "/HN- EXCEEDS MAXIMUM"
    }
  ]
}
```

Bugs detected

Amadeus Hotel

hotelName
string
(query) Search by Hotel Name. Accepts maximum 4 keywords.
Example: Hotel California
Example: Hotel Califo

REQUEST

hotelName = "Waterfront Cebu City Hotel Casino"
hotelIds = "ICPPTICA"



RESPONSE

HTTP/1.1 200 OK

Status code 200

Body

```
{  
  "data": [  
  ]  
}
```

Bugs detected

DHL

Country Code	Number of tests
AN	9
AS	1
AX	12
AZ	17
BZ	14
LA	6
LY	14
NE	11
PN	9
PS	10
PW	9
SJ	10
SO	11
UM	9
WF	9
Total	151

**The API accepted 15
country codes
not present in the
documentation**

Bugs detected

DHL

REQUEST

latitude = 91.0
longitude = 120.0



RESPONSE

HTTP/1.1 500 Internal Server Error

Status code 500

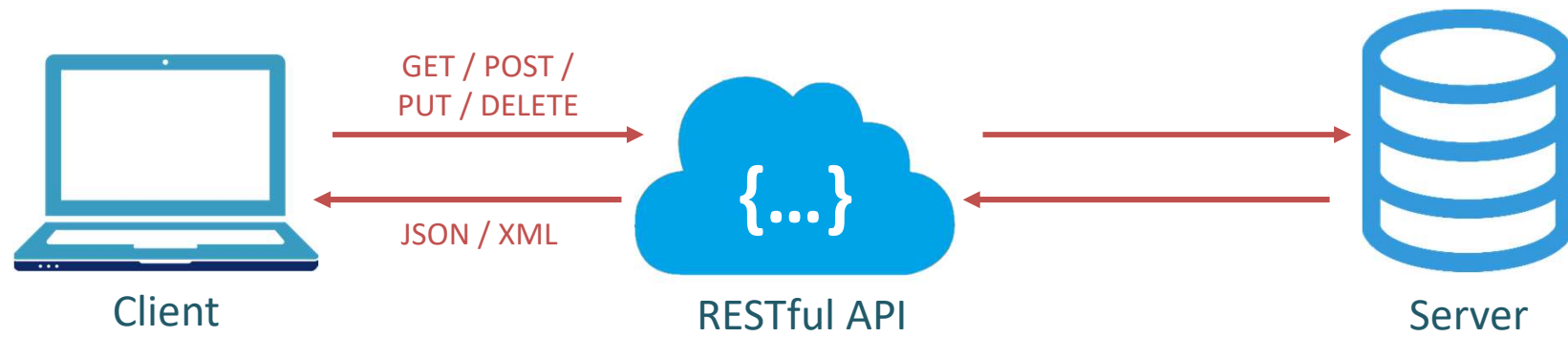
Body

```
{
  "status": 500,
  "title": "Internal Server Error",
  "detail": "The server encountered an unexpected condition that prevented it from fulfilling the request."
}
```

An out-of-range value for the parameter latitude results in a server error

Context

RESTful APIs





NETFLIX



amadeus



Context RESTful APIs



Problem

Testing web APIs requires realistic Input values



Parameters

Name	Description
countryCode * required string (query)	2-letter ISO 3166-1 alpha-2 code specifying the country.
addressLocality string (query)	Text specifying the name of the locality, for example a city.
postalCode string (query)	Text specifying the postal code for an address.
streetAddress string (query)	The street address is expressed as free form text.



Name	Description
location	Geographic area.
latitude	Latitude of the location.
longitude	Longitude of the location.

amadeus

GET

/shopping/hotel-offers FIND HOTELS.

Name	Description
cityCode	Destination City Code (or Airport Code). In case of city code, the search will be done around the city center.
hotelName	Search by Hotel Name. Accepts maximum 4 keywords.
currency	Request a specific currency. ISO currency code
lang	ISO language code

Context

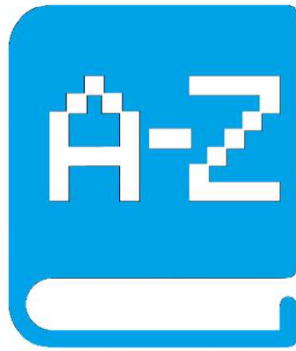
OpenAPI Specification



```
paths:
  '/find-by-address':
    get:
      operationId: findByAddress
      description: Find DHL locations based on an address.
      produces:
        - application/json
      parameters:
        - name: countryCode
          in: query
          description: 'A two-letter ISO 3166-1 alpha-2 code.'
          required: true
          type: string
        - name: postalCode
          in: query
          description: 'Postal code for an address.'
          required: false
          type: string
        - name: locationType ...
        - name: limit ...
        - name: streetAddress ...
        - name: serviceType ...
        - name: radius ...
        - name: addressLocality ...
        - name: providerType ...
      responses:
        '200':
          description: 'List of DHL Service Point locations.'
          schema:
            $ref: '#/definitions/supermodelIoLogisticsPUDOLocations'
```

Problem

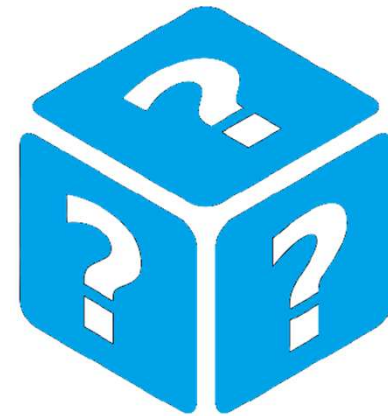
Current approaches



Data dictionaries










Default values



Fuzzing techniques









Contribution

ARTE vs SAIGEN

Features	SAIGEN (GUI)	ARTE (API SPEC)
Parameter names		
Synonyms		
Analysis of description		
Exhaustive predicate search		
Regular expressions		
Search for additional inputs		

Contribution

ARTE vs SAIGEN

Features	SAIGEN	ARTE
Parameter names		
Synonyms		
Analysis of description		
Unspecific parameter names		
Naming conventions		
Exhaustive predicate search		
Regular expressions		
Search for additional inputs		