



What if I tell you that your specs are broken

October, 10th 2019

Samuele Maci
maci@yelp.com

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN



Samuele Maci

Backend Engineer

maci@yelp.com  macisamuele

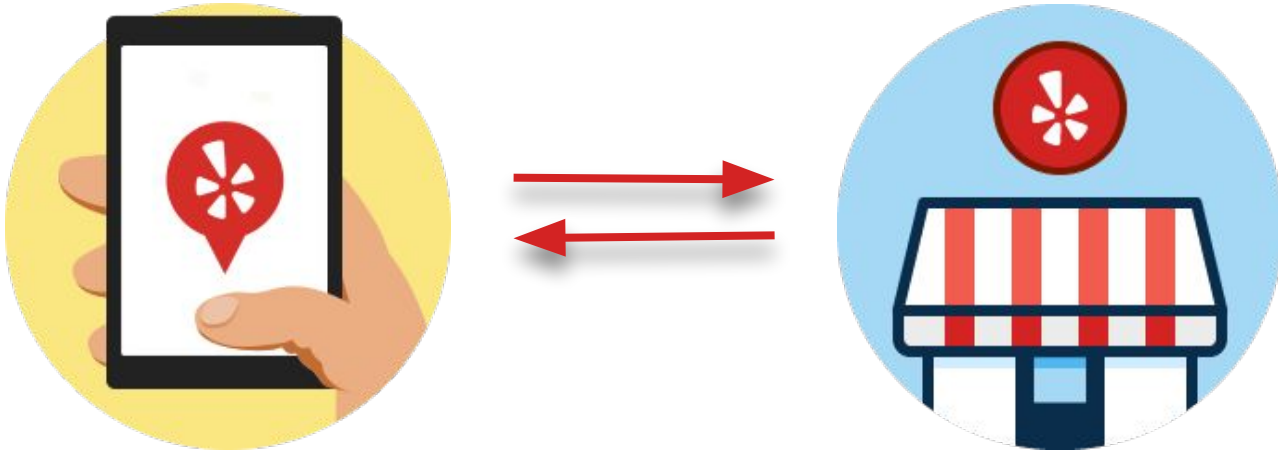
Python & Swagger Enthusiast

Maintainer of bravado, pyramid-swagger, ...



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Yelp's Mission



Connecting people with great local businesses.



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Agenda



Swagger 2.0



BookStore API

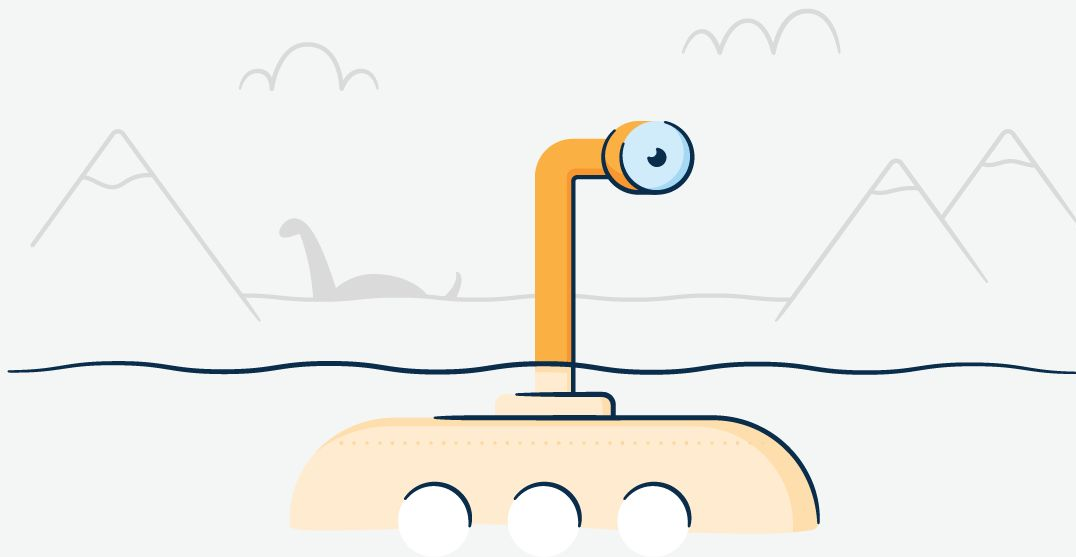


Backward Incompatibility





Swagger 2.0 Specification



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Swagger 2.0

Intro

De-facto Standard for Network APIs

Language Agnostic

Machine Readable

Extensible



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Swagger 2.0

Intro

How they look

```
# Metadata
```

```
swagger: '2.0'
```

```
info:
```

```
  title: An API title ;)
```

```
  version: 0.0.1
```

```
# Optional fields
```

```
description: A short description of the application
```

```
license:
```

```
  name: MIT
```

```
  url: https://opensource.org/licenses/MIT
```

```
termsOfService: https://tos.com/...
```

```
# Servers information (Optional but recommended values)
```

```
schemes: [http, https]
```

```
host: book-store.de.pycon.org
```

```
basePath: /
```

```
consumes: [application/json]
```

```
produces: [application/json]
```

```
securityDefinitions:
```

```
...
```



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Swagger 2.0

Intro

How they look



```
# Endpoints definitions
```

```
paths:
  /endpoint:
    get:
      responses:
        '200':
          description: Response description

      # Optional (but strongly recommended)
      schema:
        type: object
        title: ModelName
        properties:
          ...
        required:
          ...

      default:
        ...

    operationId: get_endpoint
```


WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Swagger 2.0

Intro

Specs Definition
Approaches



Manual Approach (most common @Yelp)

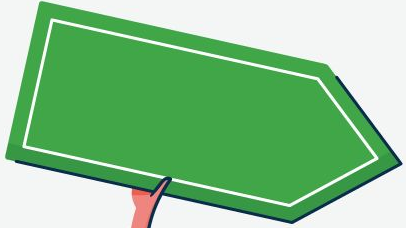
- Full control on the Specs content
- Incompatible changes are "easier" to spot



Automated approach / SpecGen

- Specs are generated from Backend code
- Incompatible changes depends on code





Example BookStore API



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

BookStore API

Context

We're defining a Web Service that allows

- to search books
- to buy books
- to borrow books
- to sell books

Authentication is required

Books modeling is not yet fully specified



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

BookStore API

Needed Endpoints

GET /v1/book/

Retrieve the book catalogue

GET /v1/book/{isbn}

Retrieve detailed book information

POST /v1/book/buy/{isbn}

Buy a book from the Store

POST /v1/book/borrow/{isbn}/{start_date}

Borrow a book from the Store

POST /v1/book/sell/{isbn}

Sell a book to the Store



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

BookStore API

Let's move fast

Define Book model

definitions:

Book:

type: object

title: Book

properties:

isbn:

description: Unique identifier of the book

type: string

title:

description: Title of the book

type: string

authors:

description: Authors of the book

items:

\$ref: '#/definitions/Author'

minItems: 1

type: array

required: [authors, isbn, title]



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

BookStore API

Let's move fast

Define Book model

```
Book ∨ {  
  authors*
```

```
  ∨ [  
    minItems: 1
```

Authors of the book

```
  Author ∨ {  
    description:
```

Detailed information of the Book Author

```
    name* string
```

Name of the author

```
  }]  
  string
```

Unique identifier of the book

```
  isbn*
```

```
  string
```

Title of the book

```
  title*
```

```
}
```



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

BookStore API

Let's move fast

Quick API Scan



book

GET	/v1/book	Retrieve the book catalogue	
GET	/v1/book/{isbn}	Retrieve book details	
POST	/v1/book/buy/{isbn}	Buy a book from the BookStore	
POST	/v1/book/borrow/{isbn} /{start_date}	Borrow a book from the BookStore	
POST	/v1/book/sell/{isbn}	Sell a book to the BookStore	


WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

BookStore API

Let's move fast


Search API - Details



GET /v1/book Retrieve the book catalogue 

Parameters Try it out

Name	Description
query <small>★ required</small> string (query)	Search a book by query <div>query - Search a book by `query`</div>

Responses Response content type **application/json** 

Code	Description
200	<div>HTTP/200 Ok</div> <div>Example Value Model</div> <div><pre>{ "books": [{ "authors": [{ "name": "string" }], "isbn": "string", "title": "string" }]}</pre></div>

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

BookStore API

Requirements Update



Search should allow filtering by release date

- Did we have release date on the model?
- Should the parameter be required or optional?

GET

/v1/book Retrieve the book catalogue

Parameters

Try it out

Name	Description
query <small>★ required</small> string (query)	Search a book by query <div>query - Search a book by `query`</div>
releaseDate string(\$date) (query)	Minimum release date <div>releaseDate - Minimum release date</div>

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

BookStore API

Requirements Update - 2nd



Allow users to search for books by genre

- Did we have genre on the model?
- Should the parameter be required or optional?

GET

/v1/book Retrieve the book catalogue

Parameters

Try it out

Name	Description
query <small>★ required</small> string (query)	Search a book by query <div>query - Search a book by `query`</div>
releaseDate string(\$date) (query)	Minimum release date <div>releaseDate - Minimum release date</div>
genre string (query)	Genre of the book (is this a biography, drama, fairytale, etc.) Available values : BIOGRAPHY, DRAMA, DAIRYTALE, HORROR <div>...</div>

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

BookStore API

Requirements Update - 3rd



Enlarge set of supported books genre

GET

/v1/book Retrieve the book catalogue

Parameters

Try it out

Name	Description
query <small>★ required</small> string (query)	Search a book by query <div>query - Search a book by `query`</div>
releaseDate string(\$date) (query)	Minimum release date <div>releaseDate - Minimum release date</div>
genre string (query)	Genre of the book (is this a biography, drama, fairytale, etc.) <i>Available values</i> : BIOGRAPHY, DAIRYTALE, DETECTIVE_STORY, DICTIONARY, DRAMA, DYSTOPIA, FANTASY, HAIKU, HORROR, MEMOIR, MUSICAL, MYSTERY, PLAY, ROMANCE, SATIRE, SCIENCE_FICTION, THRILLER, WESTERN

Backward Incompatibility



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Backward Compatibility

Definition

***Backward compatibility** is a property of a system, product, or technology that allows **interoperability** with an older **legacy** system.*



WIKIPEDIA
The Free Encyclopedia



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Backward Compatibility

When you can ignore it

You have full control on all the clients out there

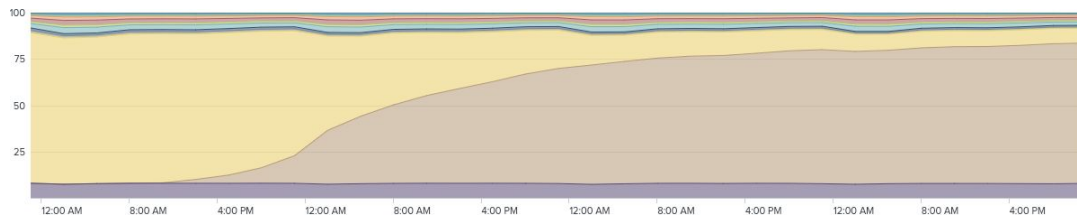
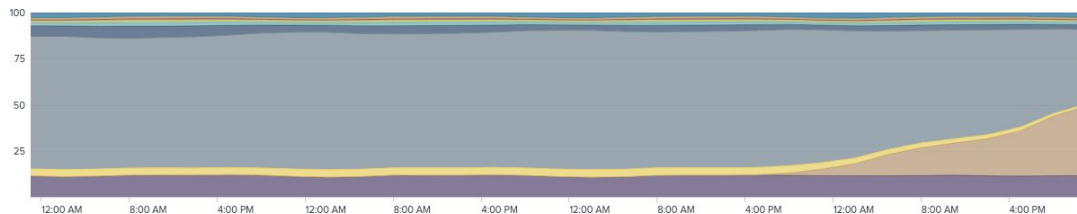
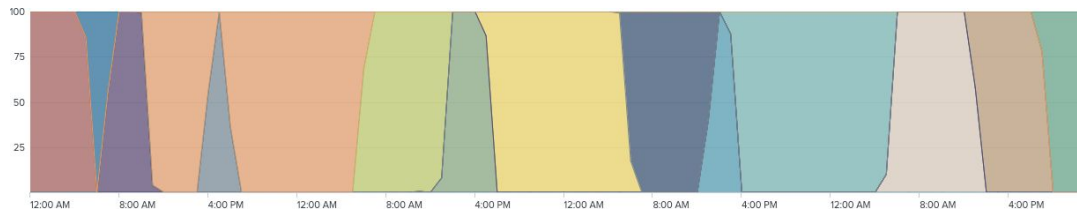
Specs changes are versioned and independent



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Backward Compatibility

Web vs APIs Used versions



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Backward Incompatible Changes

Not desirable

Impacted User Experience

Hard to identify from error rate anomalies

Tracing the incompatible change from raise of errors is not a trivial task

Usually caused by unintentional changes

Change of a parameter definition impacts a different endpoint that is reusing the definition



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Backward Incompatible Changes

How to avoid them

Define guidelines and conventions

Limits a bit the developers freedom
Increase stability and consistency of the specs

Use/Build/Integrate tools to identify them

Humans check are possible but it's time consuming
and tedious



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Yelp's Tool

`swagger-spec-compatibility`



Automatically detect safety of Swagger changes

Implemented as a static analyser / linter

It's educational

Helps developers to reduce surface risk

Provides explanation of the provided reports

It's modular and configurable

This allows you to define custom rules as well as ignoring pre-existing rules



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

How to use

```
$ swagger_spec_compatibility --help
```

```
usage: swagger_spec_compatibility [-h] {explain,info,run} ...
```

Tool for the identification of backward incompatible changes between two swagger specs.

The tool provides the following level of results:

- WARNING: the Swagger specs are technically compatible but they are likely to break known Swagger implementations
- ERROR: new Swagger spec does introduce a breaking change respect the old implementation

positional arguments:

{explain,info,run}	help for sub-command
explain	explain selected rules
info	Reports tool's information
run	run backward compatibility detection

optional arguments:

-h, --help	show this help message and exit
------------	---------------------------------



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

Swagger Specs modified three times

Have we broke the specs?

Recap

1. allow filtering by release date
2. allow filtering by genre
3. update genre list



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

> release date filtering



GET

/v1/book Retrieve the book catalogue

Parameters

Try it out

Name	Description
query <small>★ required</small> string <i>(query)</i>	Search a book by query <div>query - Search a book by `query`</div>

Responses

Response content type application/json

Code	Description
200	HTTP/200 Ok <div>Example Value Model</div> <pre>{ "books": [{ "authors": [{ "name": "string" }], "isbn": "string", "title": "string" }] }</pre>

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

> release date filtering



GET /v1/book Retrieve the book catalogue

Parameters Try it out

Name	Description
query <small>★ required</small> string (query)	Search a book by query query - Search a book by `query`
releaseDate string(\$date) (query)	Minimum release date releaseDate - Minimum release date

Responses Response content type application/json ▼

Code	Description
200	HTTP/200 Ok Example Value Model <pre>{ "books": [{ "authors": [{ "name": "string" }], "isbn": "string", "releaseDate": "2019-10-09", "title": "string" }] }</pre>

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

> release date filtering

```
$ swagger_spec_compatibility run specs_v0.yaml specs_v1.yaml
ERROR rules:
  [REQ-E001] Added Required Property in Request contract: ...
```

Exited with status_code=1



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

> release date filtering

```
$ swagger_spec_compatibility run specs_v0.yaml specs_v1.yaml
```

```
...
```

```
Exited with status_code=1
```

```
$ swagger_spec_compatibility -r=REQ-E001 explain
```

```
[REQ-E001] Added Required Property in Request contract:
```

Adding a required property to an object used in requests leads
client request to fail if the property is not present.

More info on

<https://swagger-spec-compatibility.readthedocs.io/en/latest/rules/REQ-E001.html>



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

> release date filtering



```
$ swagger_spec_compatibility run specs_v0.yaml specs_v1.yaml
```

```
...
```

```
Exited with status_code=1
```

```
$ swagger_spec_compatibility -r=REQ-E001 explain
```

```
[REQ-E001] Added Required Property in Request contract:
```

Adding a required property to an object used in requests leads client request to fail if the property is not present.

More info on

<https://swagger-spec-compatibility.readthedocs.io/en/latest/rules/REQ-E001.html>

[REQ-E001] - Added Required Property in Request contract

Rationale

Adding a required property to an object used in requests leads client request to fail if the property is not present.

Mitigation

A possible mitigation consists of adding the property as optional with an associated default value. In this case, the client requests don't fail to validate and the service can assume that the property is always set.


WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

> release date filtering

```
$ swagger_spec_compatibility run specs_v0.yaml specs_v1.yaml --json | jq .
{
  "ERROR": [
    {
      "error_code": "REQ-E001",
      "reference": "#/paths/v1/book/sell/{isbn}/post/parameters/1/schema",
      "short_name": "Added Required Property in Request contract",
      "documentation": "https://.../REQ-E001.html"
    }
  ]
}
```

POST /v1/book/sell/{isbn} Sell a book to the BookStore 

Parameters Try it out

Name	Description
isbn <small>* required</small> string (path)	ISBN of the book to sell
<div>isbn - [ISBN](https://en.wikipedia.org/wiki/Inte</div>	
body <small>* required</small> object (body)	Details of the book to sell to the BookStore
Example Value Model	
<div>Book > {...}</div>	

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

> release date filtering



Book model shared by multiple endpoints

```
title: Book
type: object
properties:
  authors:
    items:
      $ref: '#/definitions/Author'
    minItems: 1
    type: array
  isbn:
    type: string
  title:
    type: string
required: [authors, isbn, title]
```

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

> release date filtering



Book model shared by multiple endpoints

```
title: Book
type: object
properties:
  authors:
    items:
      $ref: '#/definitions/Author'
    minItems: 1
    type: array
  isbn:
    type: string
  releaseDate:
    format: date
    type: string
  title:
    type: string
required: [authors, isbn, title]
```

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

> release date filtering



Book model shared by multiple endpoints

```
title: Book
type: object
properties:
  authors:
    items:
      $ref: '#/definitions/Author'
    minItems: 1
    type: array
  isbn:
    type: string
  releaseDate:
    format: date
    type: string
  title:
    type: string
required: [authors, isbn, releaseDate, title]
```

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

> release date filtering



Book model shared by multiple endpoints

```
title: Book
type: object
properties:
  authors:
    items:
      $ref: '#/definitions/Author'
    minItems: 1
    type: array
  isbn:
    type: string
  releaseDate:
    format: date
    type: string
  title:
    type: string
required: [authors, isbn, releaseDate, title]
```

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

> genre filtering



GET

/v1/book Retrieve the book catalogue

Parameters

Try it out

Name	Description
query <small>required</small> string (query)	Search a book by query <div>query - Search a book by `query`</div>
releaseDate string(\$date) (query)	Minimum release date <div>releaseDate - Minimum release date</div>

Responses

Response content type application/json

Code	Description
200	<div>HTTP/200 Ok</div> <div>Example Value Model</div> <div><pre>{ "books": [{ "authors": [{ "name": "string" }], "genre": "BIOGRAPHY", "releaseDate": "2019-10-09", "title": "string" }] }</pre></div>

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

> genre filtering



GET /v1/book Retrieve the book catalogue 🔒

Parameters

Try it out

Name	Description
query <small>required</small> string (query)	Search a book by query <input type="text" value="query - Search a book by `query`"/>
releaseDate string(\$date) (query)	Minimum release date <input type="text" value="releaseDate - Minimum release date"/>
genre string (query)	Genre of the book (is this a biography, drama, fairytale, etc.) Available values : BIOGRAPHY, DRAMA, DAIRYTALE, HORROR <input type="text" value="--"/>

Responses

Response content type application/json

Code	Description
200	HTTP/200 Ok <div>Example Value Model</div> <pre>{ "books": [{ "authors": [{ "name": "string" }], "genre": "BIOGRAPHY", "isbn": "string", "releaseDate": "2023-10-03", "title": "string" }] }</pre>

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API
> genre filtering

```
$ swagger_spec_compatibility run specs_v1.yaml specs_v2.yaml
```

```
Exited with status_code=0
```

```
title: Book
type: object
properties:
  authors:
    items:
      $ref: '#/definitions/Author'
    minItems: 1
    type: array
  isbn:
    type: string
  releaseDate:
    format: date
    type: string
  title:
    type: string
required: [authors, isbn, releaseDate, title]
```



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API
> genre filtering

```
$ swagger_spec_compatibility run specs_v1.yaml specs_v2.yaml
```

```
Exited with status_code=0
```

```
title: Book
type: object
properties:
  authors:
    items:
      $ref: '#/definitions/Author'
    minItems: 1
    type: array
  genre:
    type: string
    enum: [BIOGRAPHY, DRAMA, ...]
  isbn:
    type: string
  releaseDate:
    format: date
    type: string
  title:
    type: string
required: [authors, isbn, releaseDate, title]
```



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API
> genre filtering



```
$ swagger_spec_compatibility run specs_v1.yaml specs_v2.yaml
```

```
Exited with status_code=0
```

```
title: Book
type: object
properties:
  authors:
    items:
      $ref: '#/definitions/Author'
    minItems: 1
    type: array
  genre:
    type: string
    enum: [BIOGRAPHY, DRAMA, ...]
  isbn:
    type: string
  releaseDate:
    format: date
    type: string
  title:
    type: string
required: [authors, isbn, releaseDate, title]
```

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

> genre list update



GET

/v1/book Retrieve the book catalogue

Parameters

Try it out

Name	Description
query <small>* required</small> string (query)	Search a book by <code>query</code> <div>query - Search a book by `query`</div>
releaseDate string(\$date) (query)	Minimum release date <div>releaseDate - Minimum release date</div>
genre string (query)	Genre of the book (is this a biography, drama, fairytale, etc.) Available values : BIOGRAPHY, DRAMA, DAIRYTALE, HORROR <div>...</div>

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

> genre list update



GET /v1/book Retrieve the book catalogue

Parameters Try it out

Name	Description
query <small>* required</small> string <i>(query)</i>	Search a book by query query - Search a book by `query`
releaseDate string(\$date) <i>(query)</i>	Minimum release date releaseDate - Minimum release date
genre string <i>(query)</i>	Genre of the book (is this a biography, drama, fairytale, etc.) <div><i>Available values</i> : BIOGRAPHY, DAIRYTALE, DETECTIVE_STORY, DICTIONARY, DRAMA, DYSTOPIA, FANTASY, HAIKU, HORROR, MEMOIR, MUSICAL, MYSTERY, PLAY, ROMANCE, SATIRE, SCIENCE_FICTION, THRILLER, WESTERN</div>

WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Tool in Action

BookStore API

> genre list update

```
$ swagger_spec_compatibility run specs_v2.yaml specs_v3.yaml --json
{
  "ERROR": [
    {
      "error_code": "RES-E003",
      "reference": "#/paths//v1/book/{isbn}/get/responses/.../genre",
      "short_name": "Added Enum value in Response contract",
      "documentation": "https://.../RES-E003.html"
    },
    { ... }
  ]
}
```

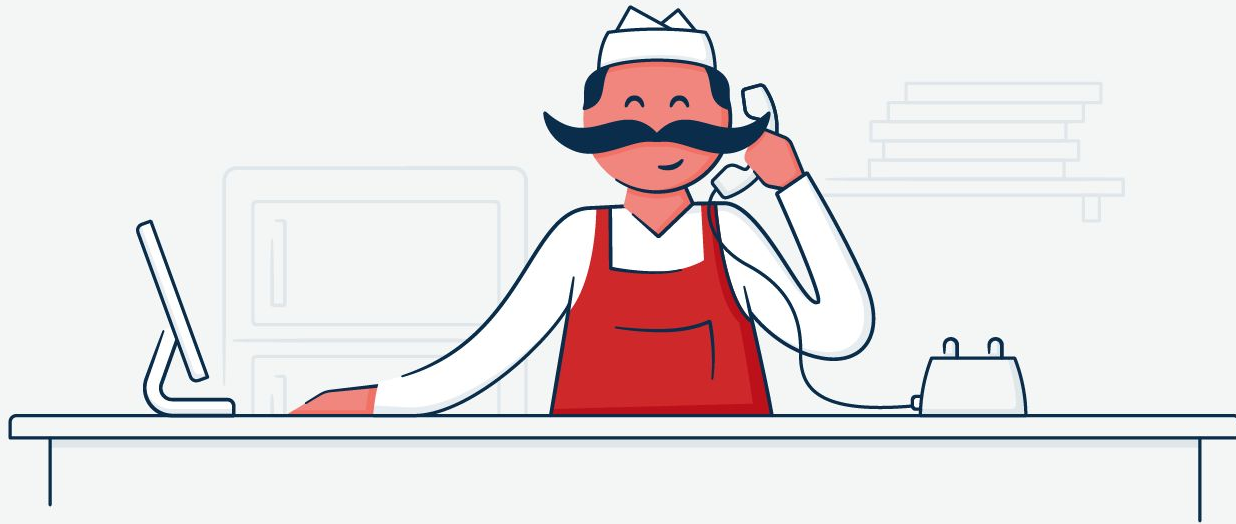
Exited with status_code=1

Mitigation

A possible mitigation consists of modifying the endpoint to get the list of enum values supported for the specific request. This way we can ensure (at the business logic level, not on Swagger) that the response will not contain enum values that are not manageable by clients using “old” Swagger spec version.



Let's recap



WHAT IF I TELL YOU THAT
YOUR SPECS ARE BROKEN

Let's recap

Swagger Spec changes are inevitable

Unrealistic to have a holistic view of the services

Backward compatibility is important

Backward compatibility should **ALWAYS** be verified






Questions/Suggestions?

maci@yelp.com

 macisamuele



Slides and Swagger Specs available at  [macisamuele/backward_incompatiblity_talk](https://github.com/macisamuele/backward_incompatiblity_talk)