False It Before You Make It: Mocking Your Way to Better HTTP APIs

Ian Zelikman
ian.zelikman@gmail.com
@izcoder

Dave Forgac
dave@forgac.com
@tylerdave
Slides, Examples, & More:
BetterAPIs.com
TERMINOLOGY
REST
REST

Standard
REST

STANDARD

ARCHITECTURAL STYLE
REST

ARCHITECTURAL STYLE

HTTP w/ CONSTRAINTS

STANDARD
REST

Standard Architectural Style
HTTP w/ Constraints
REST-inspired HTTP APIs
JSON
JSON

JavaScript Object Notation
JSON

JavaScript Object Notation

```json
{
    "things": [
        "foo",
        "bar"
    ],
    "message": "Hello, World!"
}
```
JSON Schema
```json
{
    "title": "Example Schema",
    "type": "object",
    "properties": {
        "displayName": {
            "type": "string"
        },
        "age": {
            "description": "Age in years",
            "type": "integer",
            "minimum": 0
        }
    },
    "required": ["firstName", "lastName"]
}
```
YAML

Serialization format
YAML

Serialization format

(More) Human-readable
YAML

Serialization format

(More) Human-readable

Superset of JSON
YAML

Serialization format
(More) human-readable
Superset of JSON
Broad Support
---
talk:
  title: Fake It Before You Make It
  subtitle: Mocking Your Way to Better HTTP APIs
  topics:
    - apis
    - development
  bio: >
    A talk about defining API contracts before developing implementations.
...
This Talk
Requirement
Requirement

Accept POST
Requirement

Accept POST

Return OK/validation errors
IMPLEMENTATION
IMPLEMENTATION

Accept POST
IMPLEMENTATION

Accept POST

Return OK/validation errors
“That's not really what we wanted.”
“That's not really what we wanted.”
**We provided**

```json
{
    "valid": false,
    "errors": {
        "field_1": "too short",
        "field_2": "required"
    }
}
```

**They expected**

```json
{
    "valid": false,
    "errors": [
        {
            "field": "field_1",
            "error": "too short",
        },
        {
            "error": "required",
            "field": "field_2",
        }
    ]
}
```

**The Difference**

*simplified example*
Can we do better?
Can we do better?
API Contract!
API Contract
API Contract

Consumer ↔ Provider
API Contract

Consumer ↔ Provider

Interface Specification
API Contract

Consumer ↔ Provider

Interface Specification

SLA, ToS, Limits, Pricing, etc.
API Specification
API Specification

Machine Readable
API Specification

Machine Readable Requests / Responses

Basis of Contract
Spec Formats
Spec Formats

WSDL
Spec Formats

WSDL
WADL
For REST(-like) APIs
OpenAPI Spec
(fka Swagger)
OpenAPI Spec
(fka Swagger)
JSON or YAML
OpenAPI Spec
(fka Swagger)
JSON or YAML
OpenAPI Initiative
Swagger / OpenAPI Spec 2.0 (1/3)

swagger: '2.0'
info:
  version: '1.0.0'
  title: Fake Better API
  description: >
    This is a basic example API for our talk at http://betterapis.com.
  termsOfService: http://betterapis.com/terms/
contact:
  name: API Team
  email: apiteam@betterapis.com
  url: http://betterapis.com/contact/
host: api.betterapis.com
basePath: /
securityDefinitions: {} 
schemes:
  - http 
consumes: 
  - application/json
produces:
  - application/json
paths:
/foo:
  get:
    operationId: listFoo
    produces:
      - application/json
    parameters:
      - name: limit
        in: query
        required: false
        default: 10
        type: integer
        format: int32
        description: maximum number of foo to return
    responses:
      200:
        description: List all foo
        schema:
          type: array
          items:
**Swagger / OpenAPI Spec 2.0 (3/3)**

```json
definitions:
  Foo:
    title: Foo
    description: An example type of object
    type: object
    properties:
      name:
        type: string
      score:
        type: integer
        format: int32
```

(23/43)
RAML
RAML

YAML Specification
RAML

YAML Specification
Modeling
RAML (1/3)

#%RAML 1.0

title: Fake Better API
version: 1.0.0
baseUri: http://api.betterapis.com/
baseUriParameters: {}
documentation:
- title: Fake Better API
  content: This is a basic example API for our talk at http://betterapis.com.
RAML (2/3)

types:
    Foo:
        displayName: Foo
        description: An example type of object
        type: object
        properties:
            name:
                required: false
                displayName: name
                type: string
            score:
                required: false
                displayName: score
                type: integer
                format: int32
/foo:
  get:
    displayName: listFoo
    queryParameters:
      limit:
        required: false
default: 10
displayName: limit
description: maximum number of foo to return
type: integer
format: int32
responses:
  200:
    description: List all foo
    body:
      application/json:
        displayName: response
description: List all foo
type: array
items:
API Blueprint
API Blueprint

Documentation as Specification
API BLUEPRINT

DOCUMENTATION AS SPECIFICATION

MARKDOWN + EXTENSIONS
FORMAT: 1A
HOST: http://api.betterapis.com/

# Fake Better API
This is a basic example API for our talk at http://betterapis.com.

# Foo [/foo{?limit}] 

## listFoo [GET]
+ Parameters
  + limit (number, optional) -
    
    maximum number of foo to return

  + Default: 10

+ Response 200 (application/json)

  List all foo
# Data Structures

## Foo (object)
An example type of object

### Properties

+ `name` (string, optional)
+ `score` (number, optional)
Comparison
Comparison

User-base
COMPARISON

USER-BASE

TOOLING
Comparison

User-base

Tooling

Language Support
COMPARISON

USER-BASE

TOOLING

LANGUAGE SUPPORT

RECENT EVENTS
COMPARISON

User-base
Tooling
Language Support
Recent Events
It Depends...
Defining Spec
Defining Spec

Implementation First
Defining Spec
Implementation First
Contract First
Defining Spec

Implementation First

Contract First

Integrated
DOCUMENTATION
Testing
Code Generation
Mocking
Almost Demo Time!
Almost Demo Time!

Integrated Framework
Almost Demo Time!

Integrated Framework Connexion (Python)
Almost Demo Time!

Integrated Framework
Connexion (Python)

Spec as Config:
Routing
Validation
Serialization
Fake it ➔ Make it

**Spec**
- Write / Update
- Publish
- Mock

**API**
- Implement
- Validate
Demo
Tooling
TOOLING

FRAMEWORKS
Tooling
Frameworks
Mocking
TOOLING

FRAMEWORKS

MOCKING

VALIDATION / TESTING
RESOURCES
BetterAPIs.com
FURTHER READING
Thank You!

Please leave feedback!
"Rate This Session" on schedule listing

Ian
ian.zelikman@gmail.com
@izcoder

Dave
dave@forgac.com
@tylerdave