Introduction to OpenAPI

Lorna Mitchell, Nexmo

APIs are the engine of modern software development

API descriptions power up our API workflows

Spec-First API Design

New APIs or Existing Ones?

New APIs or Existing Ones?

Yes!

API Description Languages

- **API Blueprint** from Apiary
- **RAML** from Mulesoft (who also now support OpenAPI)
- **OpenAPI** is an open standard, and is used in this talk
    - "The standard formerly known as Swagger"

OpenAPI Descriptions

WARNING may contain YAML

MetaData

```
openapi: 3.0.0
servers:
  - url: 'https://api.nexmo.com/ni'
info:
  title: Number Insight API
  version: 1.0.4
  description: Nexmo's Number Insight API delivers real-time intelligence ...
  contact:
    name: Nexmo DevRel
    email: devrel@nexmo.com
    url: 'https://developer.nexmo.com/'
  termsOfService: 'https://www.nexmo.com/terms-of-use'
  license:
    name: 'The MIT License (MIT)'
    url: 'https://opensource.org/licenses/MIT'
```

Endpoints in OpenAPI Spec

```
paths:
  '/basic/{format}':
    get:
      operationId: getNumberInsightBasic

  '/standard/{format}':
    get:
      operationId: getNumberInsightStandard

  '/advanced/async/{format}':
    get:
      operationId: getNumberInsightAsync

  '/advanced/{format}':
    get:
      operationId: getNumberInsightAdvanced
```

Parameters and Responses

```yaml
parameters:
  - $ref: '#/components/parameters/format'
  - $ref: '#/components/parameters/number'
  - $ref: '#/components/parameters/country'

responses:
  '200':
    description: OK
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/niResponseJsonBasic'
      text/xml:
        schema:
          $ref: '#/components/schemas/niResponseXmlBasic'
```

OpenAPI $ref Feature

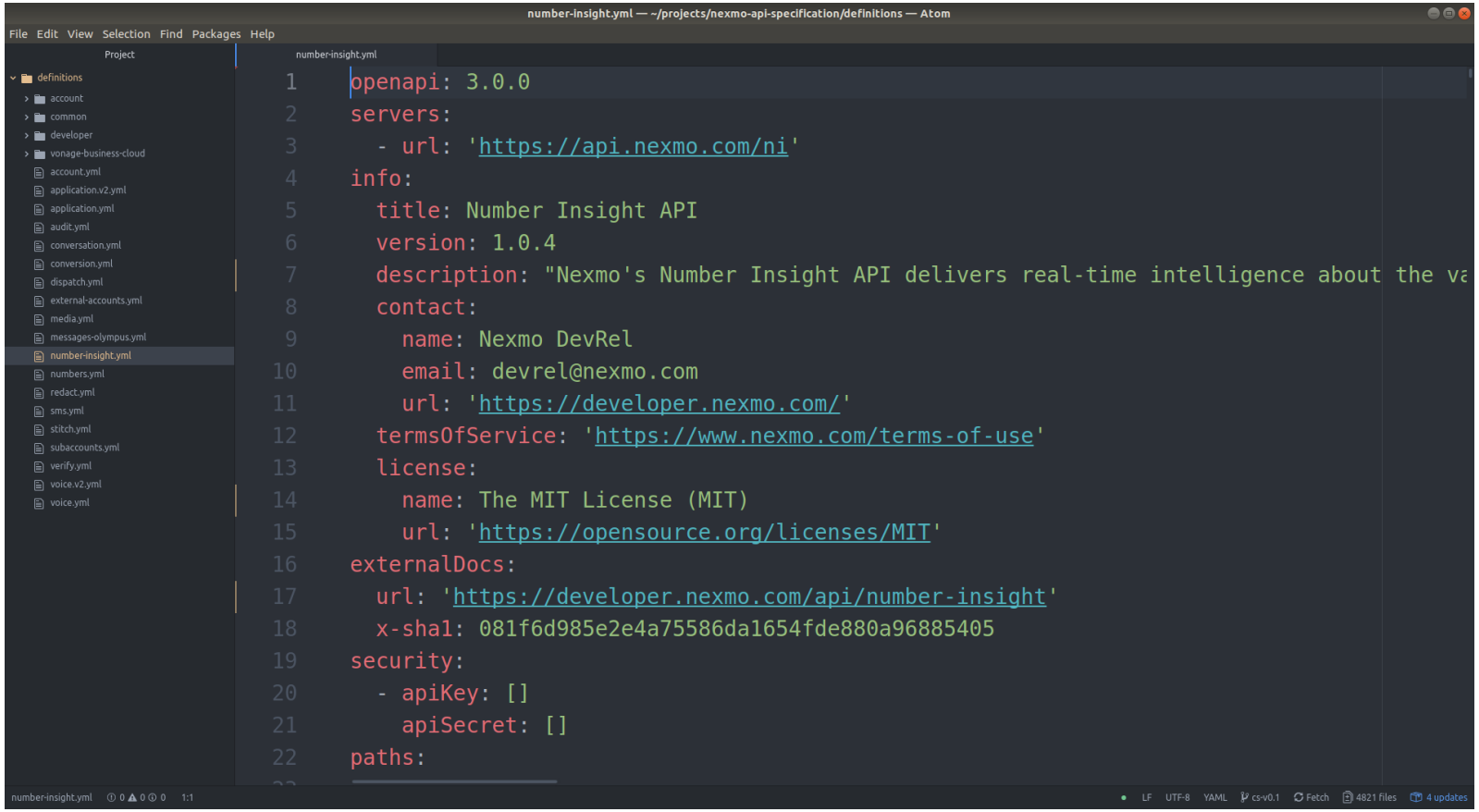
(Re)Use any block. For example the format parameter:

```yaml
components:
  parameters:
    format:
      name: format
      in: path
      required: true
      description: 'The format of the response'
      example: json
      schema:
        type: string
        enum:
          - 'json'
          - 'xml'
```

That's a lot of YAML ...

How to Edit OpenAPI Specs

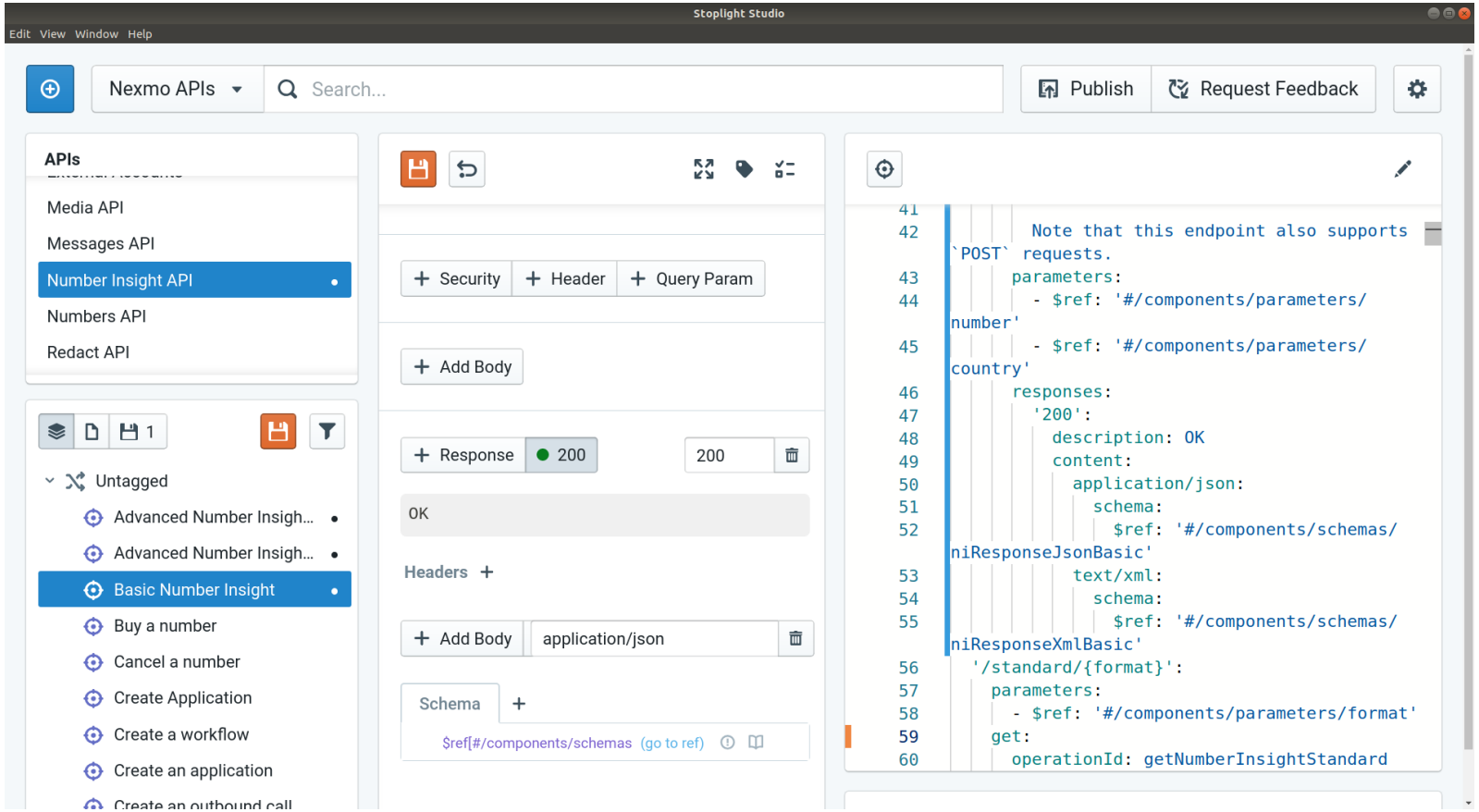
Option 1: your usual editor (this is https://atom.io)



Text-Based API Descriptions

How to Edit OpenAPI Specs

Option 2: specialist tool https://stoplight.io/studio/



Validate and Check Style

Spectral https://github.com/stoplightio/spectral

```
/home/lorna/specs/number-insight.yml
  28:9  warning  operation-tags  Operation should have non-empty `tags` array.
  51:9  warning  operation-tags  Operation should have non-empty `tags` array.
  75:9  warning  operation-tags  Operation should have non-empty `tags` array.
 119:9  warning  operation-tags  Operation should have non-empty `tags` array.

✖ 4 problems (0 errors, 4 warnings, 0 infos)
```

Pick-and-mix the rules, and sprinkle in a few of your own!

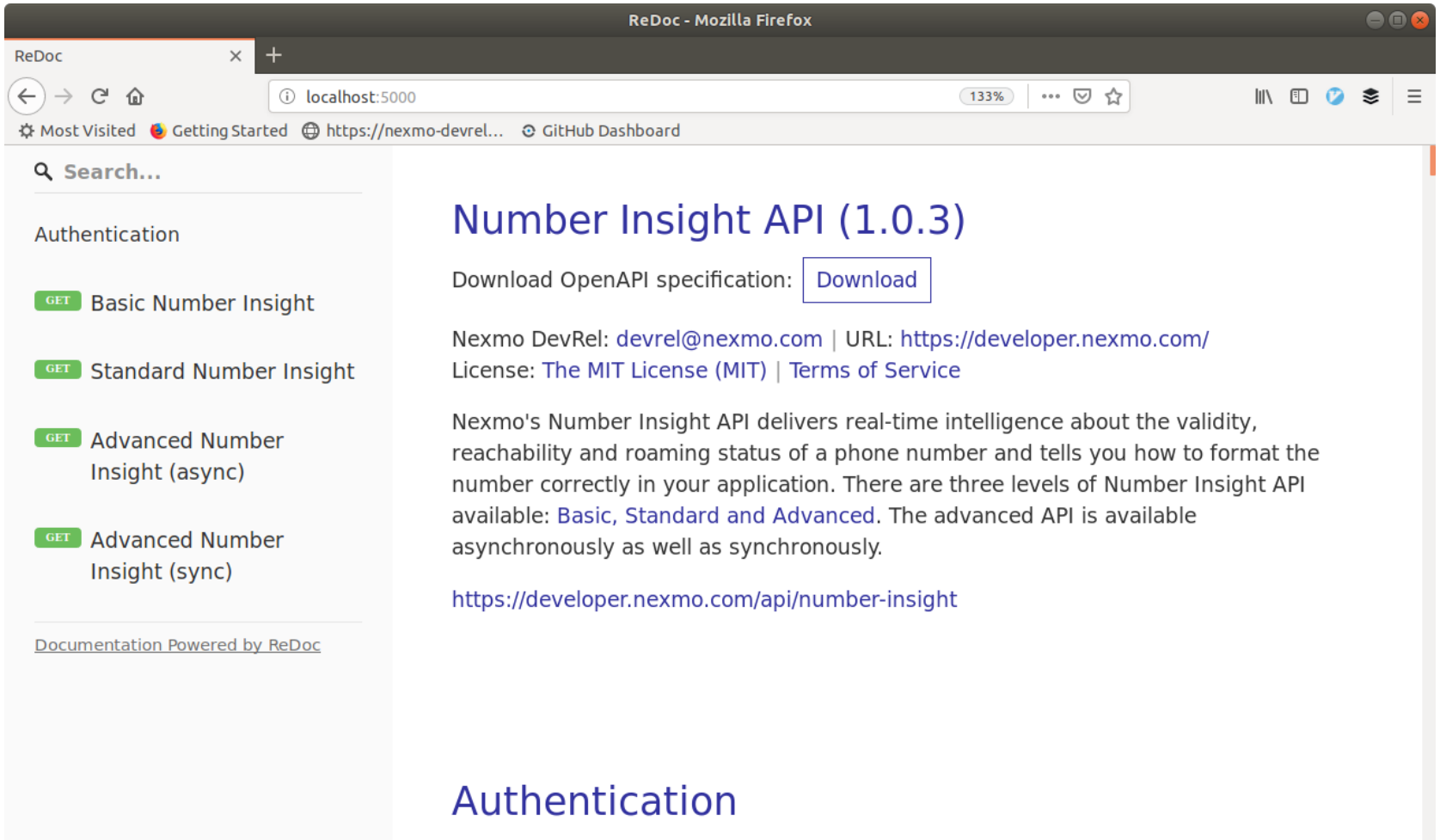
Even the machines can understand ... now what?

Generate Documentation

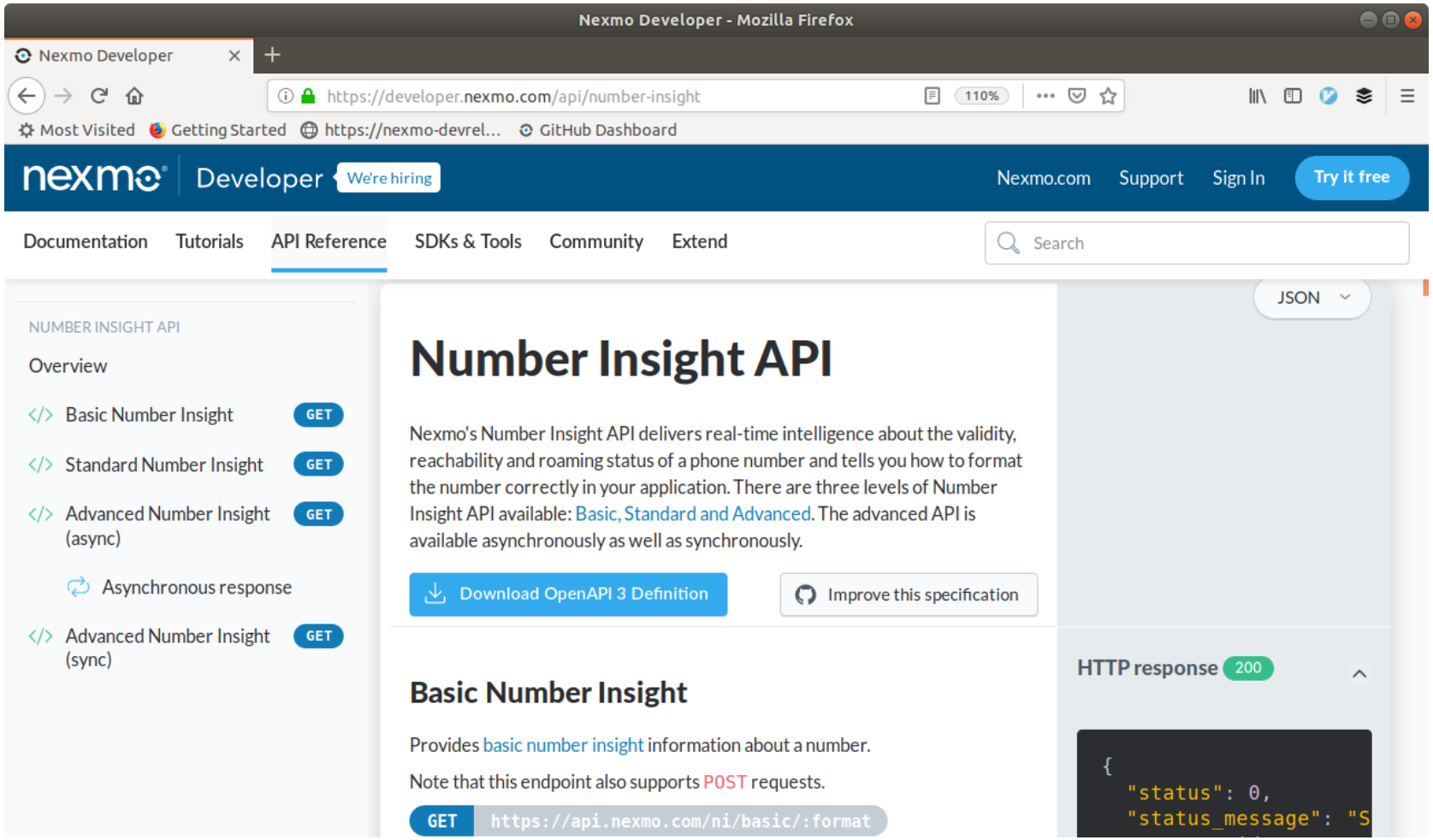
Use your OpenAPI spec as the source for your API reference documentation

- A choice of tools
- Separate content and presentation
- The spec supports and encourages examples

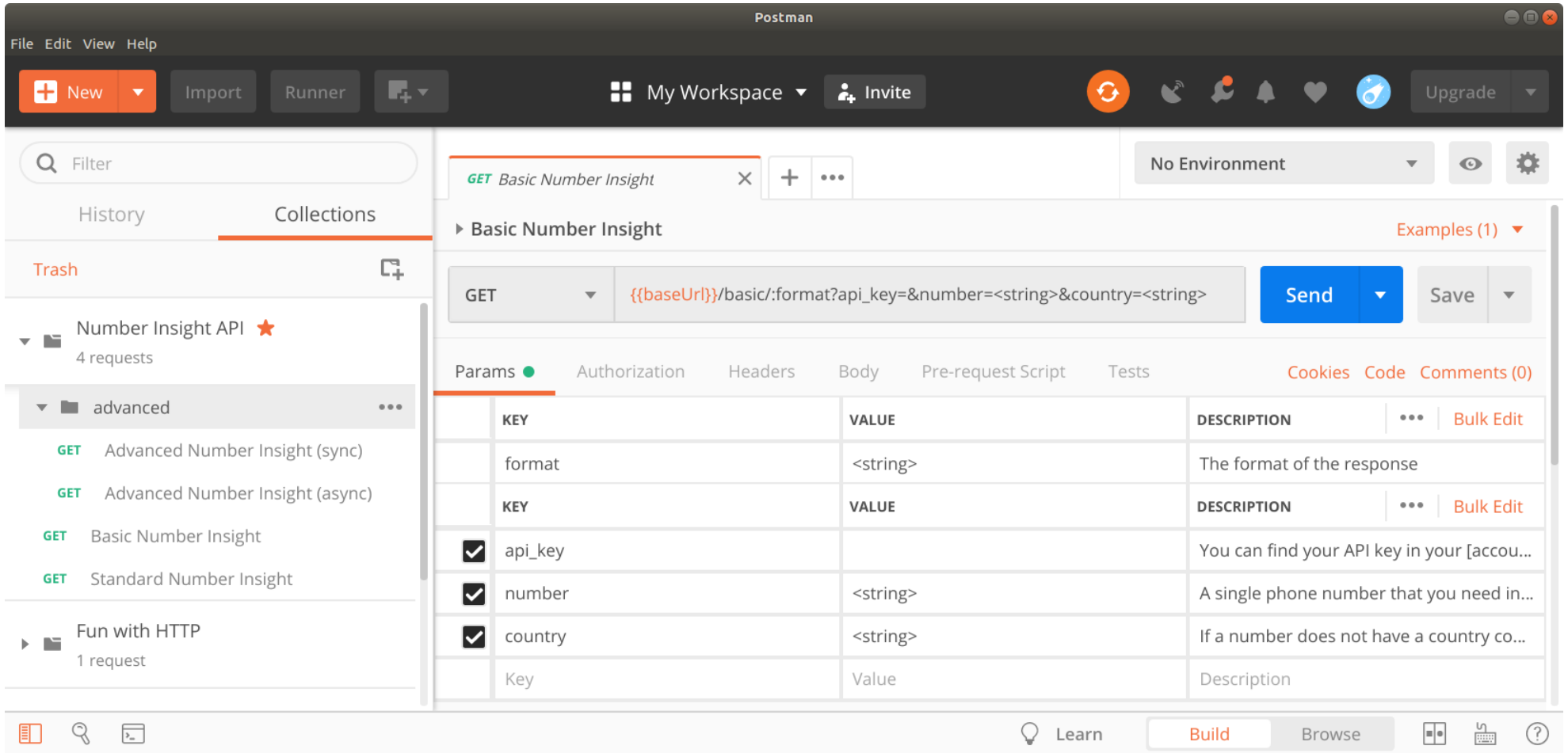
Docs Example: ReDoc



Docs Example: Nexmo



Explore APIs with Postman



OpenAPI Spec To Mock Server

Prism is a mock server https://stoplight.io/prism/

```
[HTTP SERVER] get /basic/json i  info      Request received
    [NEGOTIATOR] i  info      Request contains an accept header: */*
    [VALIDATOR] ✔  success   The request passed the validation rules. Looking for the
best response
    [NEGOTIATOR] ✔  success   Found a compatible content for */*
    [NEGOTIATOR] ✔  success   Responding with the requested status code 200
[HTTP SERVER] get /standard/json i  info      Request received
    [NEGOTIATOR] i  info      Request contains an accept header: */*
    [VALIDATOR] ✔  success   The request passed the validation rules. Looking for the
best response
    [NEGOTIATOR] ✔  success   Found a compatible content for */*
    [NEGOTIATOR] ✔  success   Responding with the requested status code 200
```

Generated Code Libraries

This example is from OpenAPI Generator
https://github.com/OpenAPITools/openapi-generator

```
docker run --rm -v ${PWD}:/local \
openapitools/openapi-generator-cli generate
-i number-insight.yml -g php -o /local/out/php
```

Generated Code Libraries

To use it:

```php
require_once('out/php/vendor/autoload.php');
// copy code from README, set API key and secret

$apiInstance = new OpenAPI\Client\Api\DefaultApi(
    new GuzzleHttp\Client(), $config);
$format = "json";
$number = "447700900000";
try {
    $result = $apiInstance->getNumberInsightBasic($format, $number);
    print_r($result);
} catch (Exception $e) {
    echo 'Exception when calling DefaultApi->getNumberInsightBasic';
}
```

Finding Tools

https://openapi.tools - a community listing

Tool Types

We've organised everything into categories so you can jump to the section you're interested in.

- **Converters:** Various tools to convert to and from OpenAPI standards. Useful for working with various API formats.
- **Data Validators:** Tools to help you validate your API data.
- **Documentation:** Generators to help build API documentation based off a variety of formats and standards.
- **DSL:** Domain Specific Language (DSL) for Open API
- **Text Editors:** Text editors to help build API docs.
- **GUI Editors:** Visual editors to help build API docs.
- **Mock Servers:** Services and platforms to speed up the testing and development of your APIs.
- **Schema Validators:** Validate your API schema quickly and easily with these tools.
- **Security:** By poking around your OpenAPI specification, some tools can look out for attack vectors you might not have noticed.
- **SDK Generators:** Quickly generate documentation, descriptions, and clients by creating a SDK.
- **Server Implementations:** Easily create and implement resources and routes for your APIs.
- **Miscellaneous:** Tools to load documents and resolve references.
- **Parsers:** Serialize and migrate API docs between versions while validating specs and docs.
- **Testing:** Quickly execute API request and validate responses on the fly through command line or GUI interfaces.

OpenAPI: Brave New World

Resources

- https://developer.nexmo.com
- https://openapis.org
- https://stoplight.io (Studio, Prism and Spectral)
- https://github.com/Nexmo/nexmo-oas-renderer
- https://github.com/Redocly/redoc
- https://openapi.tools

goto; copenhagen

GOTO Copenhagen 2019

Conference Nov. 18 - 20

Please

Remember to rate this session

Thank you!

Follow us @gotocph

@lornajane