

The background of the slide is a light blue technical drawing or blueprint. It features various geometric shapes, including circles, lines, and gears. Some areas are shaded with diagonal lines, and there are numerical annotations like '22.244', '26', and '98'. The overall aesthetic is that of a mechanical or engineering drawing.

Introduction to OpenAPI

Lorna Mitchell, Nexmo

The background is a light blue technical drawing or blueprint. It features various mechanical sketches, including gears of different sizes, circles, and lines. Some parts are shaded with diagonal hatching. There are also some numbers and arrows scattered throughout the drawing, such as '22.244' and '98'.

**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

parameters:

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

responses:

'200':

description: OK

content:

application/json:

schema:

\$ref: '#/components/schemas/nlResponseJsonBasic'

text/xml:

schema:

\$ref: '#/components/schemas/nlResponseXmlBasic'



OpenAPI \$ref Feature

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

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

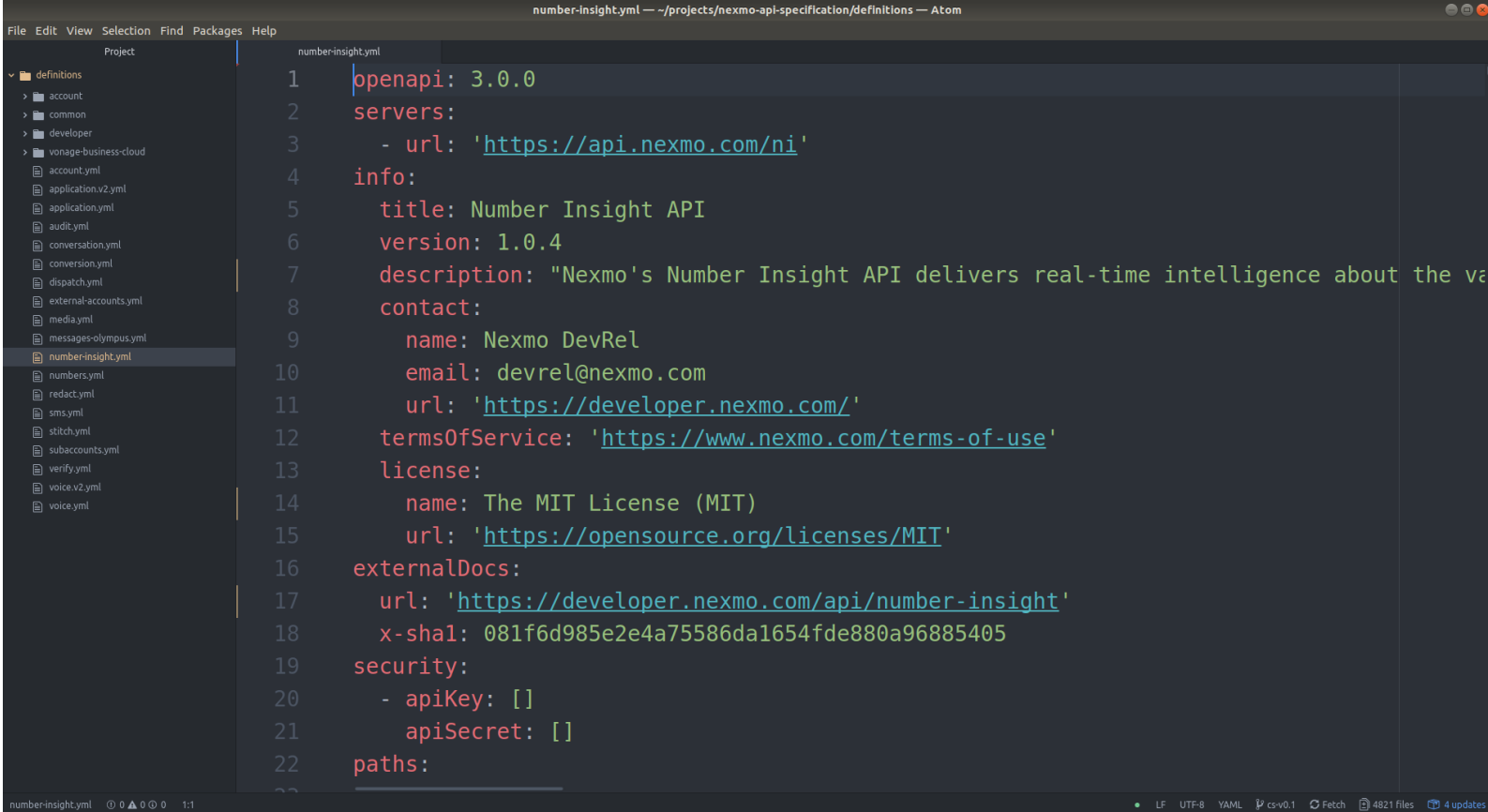


The background is a complex technical drawing in light blue ink on a white background. It features several large gears of varying sizes, some with teeth clearly defined. There are numerous overlapping circles, some with diagonal hatching. A grid of dashed lines is visible. Various arrows and dimension lines are scattered throughout, with some numerical values like '22.244', '26', '98', and '1313' written in small text. The overall appearance is that of a detailed engineering or architectural sketch.

That's a lot of YAML ...

How to Edit OpenAPI Specs

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



```
number-insight.yml -- ~/projects/nexmo-api-specification/definitions — Atom
File Edit View Selection Find Packages Help
Project
  definitions
  account
  common
  developer
  wonage-business-cloud
  account.yml
  application.v2.yml
  application.yml
  audit.yml
  conversation.yml
  conversion.yml
  dispatch.yml
  external-accounts.yml
  media.yml
  messages-olympus.yml
  number-insight.yml
  numbers.yml
  redact.yml
  sms.yml
  stitch.yml
  subaccounts.yml
  verify.yml
  voice.v2.yml
  voice.yml

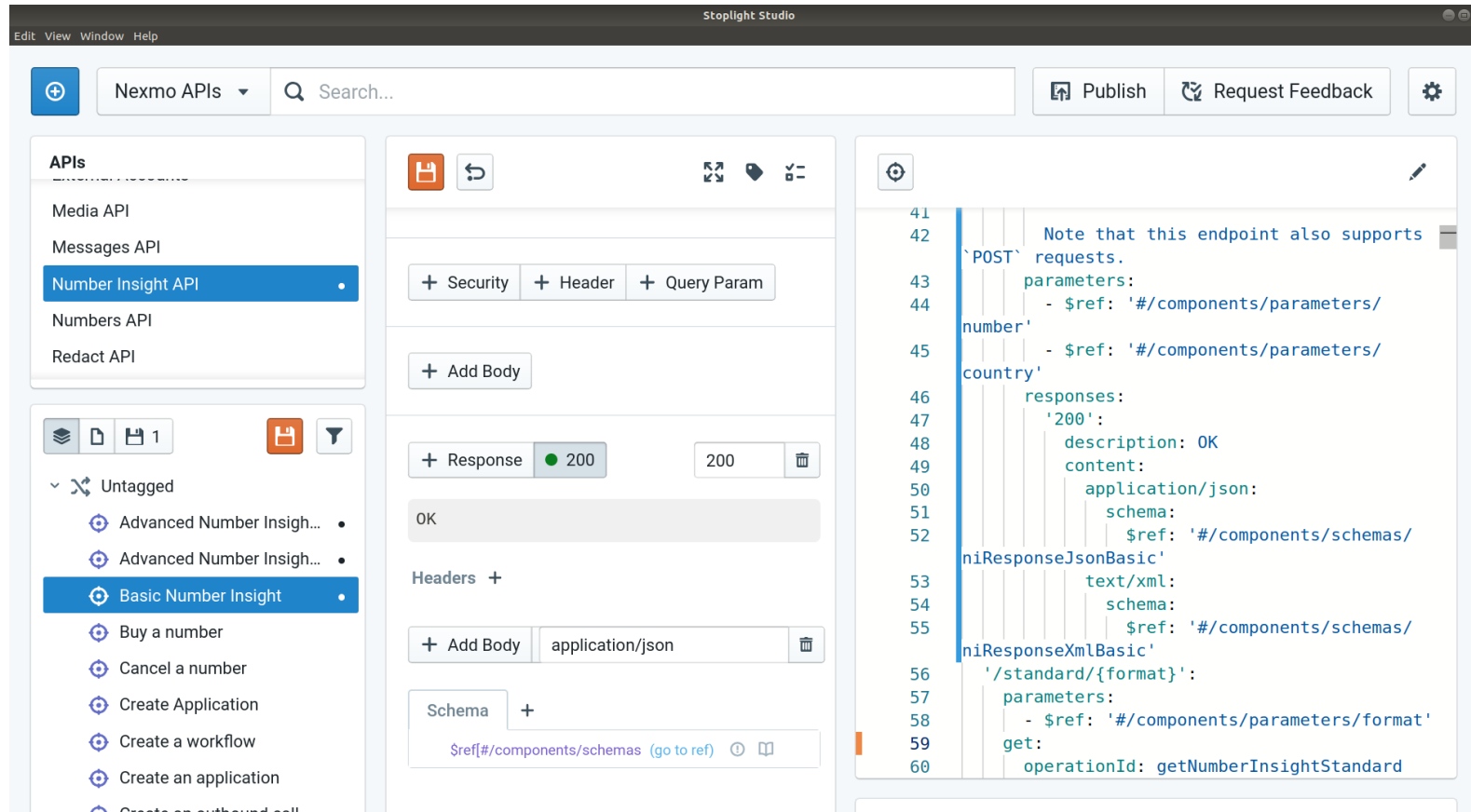
1  openapi: 3.0.0
2  servers:
3    - url: 'https://api.nexmo.com/ni'
4  info:
5    title: Number Insight API
6    version: 1.0.4
7    description: "Nexmo's Number Insight API delivers real-time intelligence about the va
8    contact:
9      name: Nexmo DevRel
10     email: devrel@nexmo.com
11     url: 'https://developer.nexmo.com/'
12    termsOfService: 'https://www.nexmo.com/terms-of-use'
13    license:
14      name: The MIT License (MIT)
15      url: 'https://opensource.org/licenses/MIT'
16  externalDocs:
17    url: 'https://developer.nexmo.com/api/number-insight'
18    x-sha1: 081f6d985e2e4a75586da1654fde880a96885405
19  security:
20    - apiKey: []
21      apiSecret: []
22  paths:
```

Text-Based API Descriptions



How to Edit OpenAPI Specs

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



The screenshot displays the Stoplight Studio interface for editing an OpenAPI specification. The top navigation bar includes 'Edit', 'View', 'Window', and 'Help' menus, along with a search bar and buttons for 'Publish', 'Request Feedback', and settings. The main workspace is divided into three panels:

- Left Panel (APIs):** A list of APIs under the 'Nexmo APIs' group, including 'Media API', 'Messages API', 'Number Insight API' (selected), 'Numbers API', and 'Redact API'. Below this is a section for 'Untagged' items, listing various actions like 'Advanced Number Insight', 'Basic Number Insight', 'Buy a number', etc.
- Middle Panel (Configuration):** A configuration area for the selected API. It includes buttons for '+ Security', '+ Header', '+ Query Param', '+ Add Body', '+ Response' (with a '200' status code), and '+ Add Body' (with 'application/json' content type). A 'Schema' section is also visible with a reference to '#/components/schemas'.
- Right Panel (Code Editor):** A code editor showing the OpenAPI specification in JSON format. The visible code includes a note about supporting POST requests, parameters for 'number' and 'country', and a response definition for '200' with 'application/json' and 'text/xml' content types. The response schema is referenced as '#/components/schemas/niResponseJsonBasic' and '#/components/schemas/niResponseXmlBasic'. The operation is defined as 'get' with 'operationId: getNumberInsightStandard'.

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!

The background is a complex technical drawing in light blue. It features several large gears of different sizes, some with hatched areas representing cross-sections. There are numerous straight lines, some solid and some dashed, forming a grid-like structure. Arrows indicate directions of movement or flow. Some numbers are scattered throughout the drawing, such as '22.244', '26', '98', and '1313'.

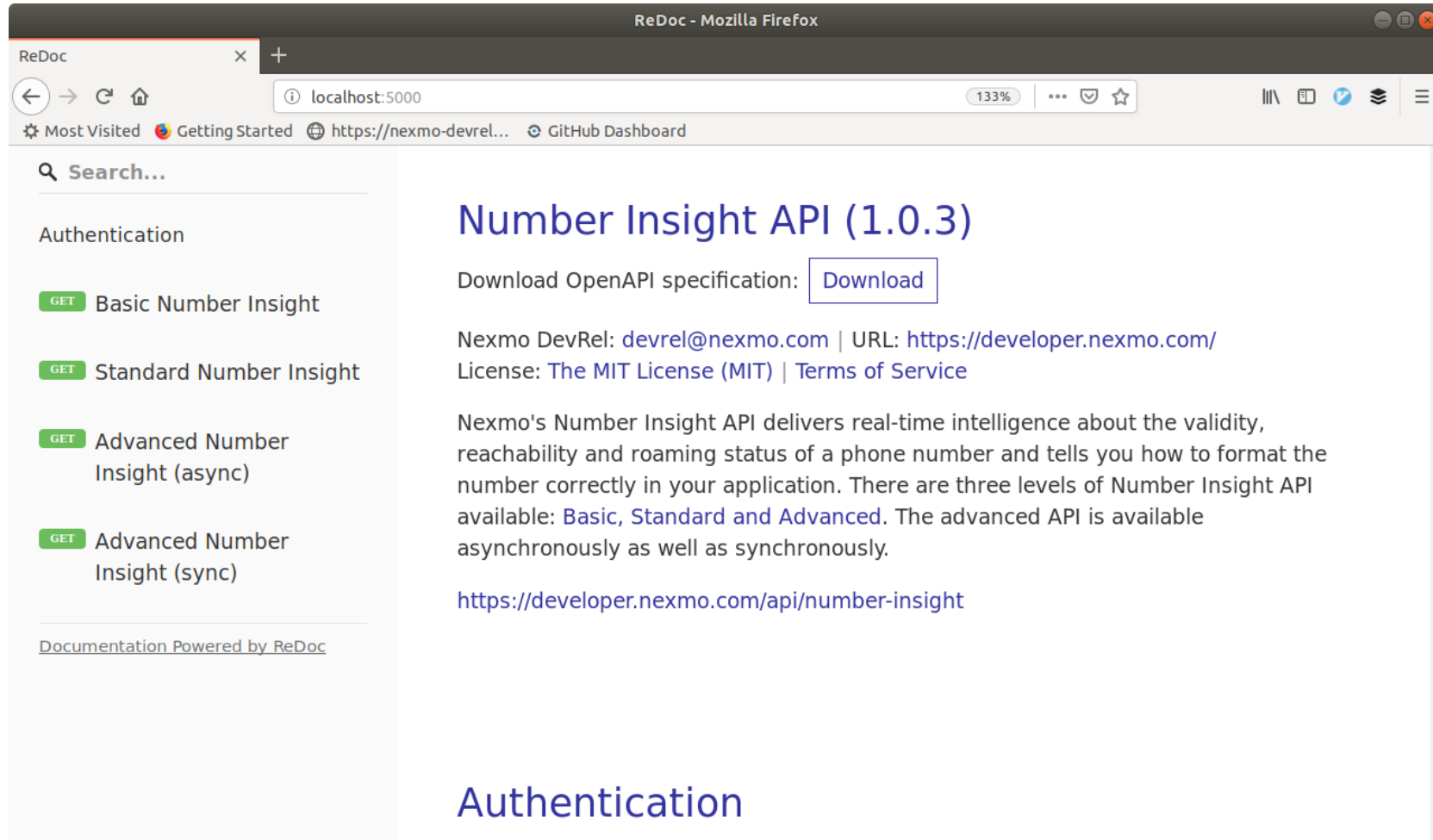
**Even the machines can understand ...
now what?**

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



The screenshot shows a web browser window titled "ReDoc - Mozilla Firefox" displaying the ReDoc API documentation for the "Number Insight API (1.0.3)". The browser address bar shows "localhost:5000". The page layout includes a left sidebar with a search bar and a list of API endpoints under the "Authentication" section. The main content area displays the API title, a "Download" button for the OpenAPI specification, contact information for Nexmo DevRel, the license (MIT), and a brief description of the API's functionality. A link to the API endpoint is also provided.

ReDoc

localhost:5000

Search...

Authentication

- GET Basic Number Insight
- GET Standard Number Insight
- GET Advanced Number Insight (async)
- GET Advanced Number Insight (sync)

Documentation Powered by ReDoc

Number Insight API (1.0.3)

Download OpenAPI specification: [Download](#)

Nexmo DevRel: devrel@nexmo.com | URL: <https://developer.nexmo.com/>
License: [The MIT License \(MIT\)](#) | [Terms of Service](#)

Nexmo's Number Insight API delivers real-time intelligence about the validity, reachability and roaming status of a phone number and tells you how to format the number correctly in your application. There are three levels of Number Insight API available: [Basic](#), [Standard](#) and [Advanced](#). The advanced API is available asynchronously as well as synchronously.

<https://developer.nexmo.com/api/number-insight>

Authentication

Docs Example: Nexmo

The screenshot shows a web browser window titled "Nexmo Developer - Mozilla Firefox" displaying the Nexmo Developer API documentation. The browser address bar shows the URL `https://developer.nexmo.com/api/number-insight`. The page header includes the Nexmo logo, the word "Developer", a "We're hiring" button, and navigation links for "Nexmo.com", "Support", "Sign In", and "Try It free". Below the header is a navigation menu with "Documentation", "Tutorials", "API Reference" (which is underlined), "SDKs & Tools", "Community", and "Extend". A search bar is located to the right of the navigation menu.

The main content area is titled "NUMBER INSIGHT API" and "Overview". On the left side, there is a list of API endpoints with their methods:

- `</>` Basic Number Insight **GET**
- `</>` Standard Number Insight **GET**
- `</>` Advanced Number Insight (async) **GET**
- `↻` Asynchronous response
- `</>` Advanced Number Insight (sync) **GET**

The main content area features a large heading "Number Insight API" followed by a paragraph: "Nexmo's Number Insight API delivers real-time intelligence about the validity, reachability and roaming status of a phone number and tells you how to format the number correctly in your application. There are three levels of Number Insight API available: **Basic**, **Standard** and **Advanced**. The advanced API is available asynchronously as well as synchronously." Below this text are two buttons: "Download OpenAPI 3 Definition" and "Improve this specification".

Below the main text is a section titled "Basic Number Insight" with the description: "Provides **basic number insight** information about a number. Note that this endpoint also supports **POST** requests." Below this is a **GET** button with the URL `https://api.nexmo.com/ni/basic/:format`.

On the right side of the page, there is a "JSON" dropdown menu and a section titled "HTTP response 200" which displays a JSON object:

```
{
  "status": 0,
  "status_message": "S"
```

Explore APIs with Postman

The screenshot displays the Postman application interface. The top menu bar includes 'File', 'Edit', 'View', and 'Help'. Below the menu is a toolbar with buttons for 'New', 'Import', 'Runner', and 'My Workspace'. The main workspace shows a REST client configuration for a GET request to 'Basic Number Insight'. The URL is `{{baseUrl}}/basic/:format?api_key=&number=<string>&country=<string>`. The 'Params' tab is active, showing a table of query parameters.

KEY	VALUE	DESCRIPTION
format	<string>	The format of the response
api_key		You can find your API key in your [accou...]
number	<string>	A single phone number that you need in...
country	<string>	If a number does not have a country co...
Key	Value	Description

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:

```
1 require_once('out/php/vendor/autoload.php');
2 // copy code from README, set API key and secret
3
4 $apiInstance = new OpenAPI\Client\Api\DefaultApi(
5     new GuzzleHttp\Client(), $config);
6 $format = "json";
7 $number = "447700900000";
8 try {
9     $result = $apiInstance->getNumberInsightBasic($format, $number);
10    print_r($result);
11 } catch (Exception $e) {
12     echo 'Exception when calling DefaultApi->getNumberInsightBasic';
13 }
```


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.



The background is a light blue technical drawing or blueprint. It features various mechanical sketches, including gears of different sizes, circles, and rectangular shapes. Some areas are filled with diagonal hatching. There are also some numerical annotations like '22.244', '26', '98', and '1313'. The overall style is that of a hand-drawn engineering plan.

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>



Please

**Remember to
rate this session**

Thank you!

