



Consuming REST APIs in APL with OpenAPI

Holden Hoover

holden@dyalog.com

About Me

~~New~~ APLer

Forge

UWaterloo Computer Engineering

Co-Op Student/Intern @ Dyalog Ltd



Web Services

Expose service to web

REST Services

Expose service to web

REST mechanism

REST Services

Expose service to web

RES *Representational State Transfer (REST)*
architectural style

REST Services

Expose service to web

REST mechanism

“Design style”

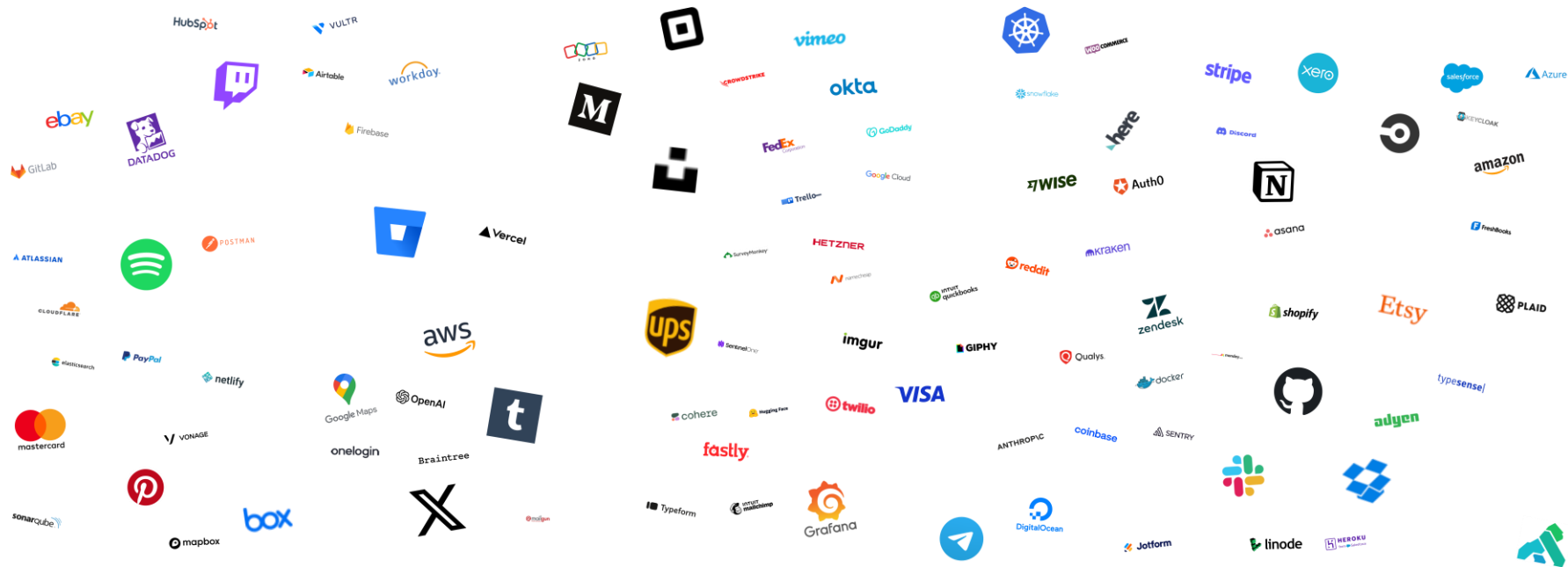
Standard HTTP methods

“Stateless”

Other

Why should I care?

Why should I care?



Why should I care?



Why should I care?



Why should I care?

The Stripe logo, consisting of the word "stripe" in a bold, blue, sans-serif font.

REST Interfaces: Now

We have some

But...

A ***lot*** of manual labour

Changes are **hard**

We want more!

What can we do?

Introducing: OpenAPI*

Standardised description of REST API

(Machine ^ Human) readable

Automation

Collaboration (Frontend+Backend)

Iteration

** Formerly known as Swagger*

OpenAPI... What?

Big long YAML/JSON file

Defines every API “operation”

Inputs and Outputs

Documentation

Much more...

```
openapi: 3.0.0
info:
  title: Small User API
  version: 1.0.0
  description: A simple API to get user information
paths:
  /users/{userId}:
    get:
      summary: Get user by ID
      parameters:
        - in: path
          name: userId
          required: true
          schema:
            type: integer
      responses:
        '200':
          description: OK
          content:
            application/json:
              schema:
                type: object
                properties:
                  id:
                    type: integer
                  name:
```



+



Platform Solutions Resources Open Source Enterprise Pricing

Search or jump to... Sign in Sign up

Dyalog / OpenAPI Public

Notifications Fork 0 Star 0

<> Code Issues 4 Pull requests Actions Projects Security and quality Insights

main Branches Tags Go to file Code

Bombardier-C-Kram Various Fixes 61 Commits e649f22 · 2 months ago

.github/workflows

Build on PR too.

2 months ago

dev-docs

Docs overhaul

2 months ago

docs

Improve examples

2 months ago

src

Various Fixes

2 months ago

.gitattributes

Mark scriban templates as APL

3 months ago

.gitignore

refactor

2 months ago

.gitmodules

documentation-styles -> documentation-assets

3 months ago

LICENSE

Initial Skeleton

3 months ago

README.md

Clean up readme

2 months ago

THIRD_PARTY_NOTICES.txt

Various Fixes

2 months ago

mkdocs.yml

Various Fixes

2 months ago

README MIT license

Dyalog APL OpenAPI Client Generator

Generates Dyalog APL client code from OpenAPI 3.0 specifications.

Release

passing

This project is in active development and may not yet be fully functional with the entirety of the OpenAPI specification. Please raise a GitHub issue for any discovered bugs or feature requests.

About

Dyalog APL OpenAPI Client Generator

[dyalog.github.io/OpenAPI/](#)

Readme

MIT license

Activity

Custom properties

0 stars

0 watching

0 forks

Report repository

Releases 2

Release v0.0.2 Latest

on Feb 27

+ 1 release

Packages

No packages published

Contributors 1

Bombardier-C-Kram

Holden Hoover

Languages

APL 54.4%

C# 45.6%

16

Consuming REST APIs in APL with OpenAPI

DYNA26

OpenAPI APL Client Generation

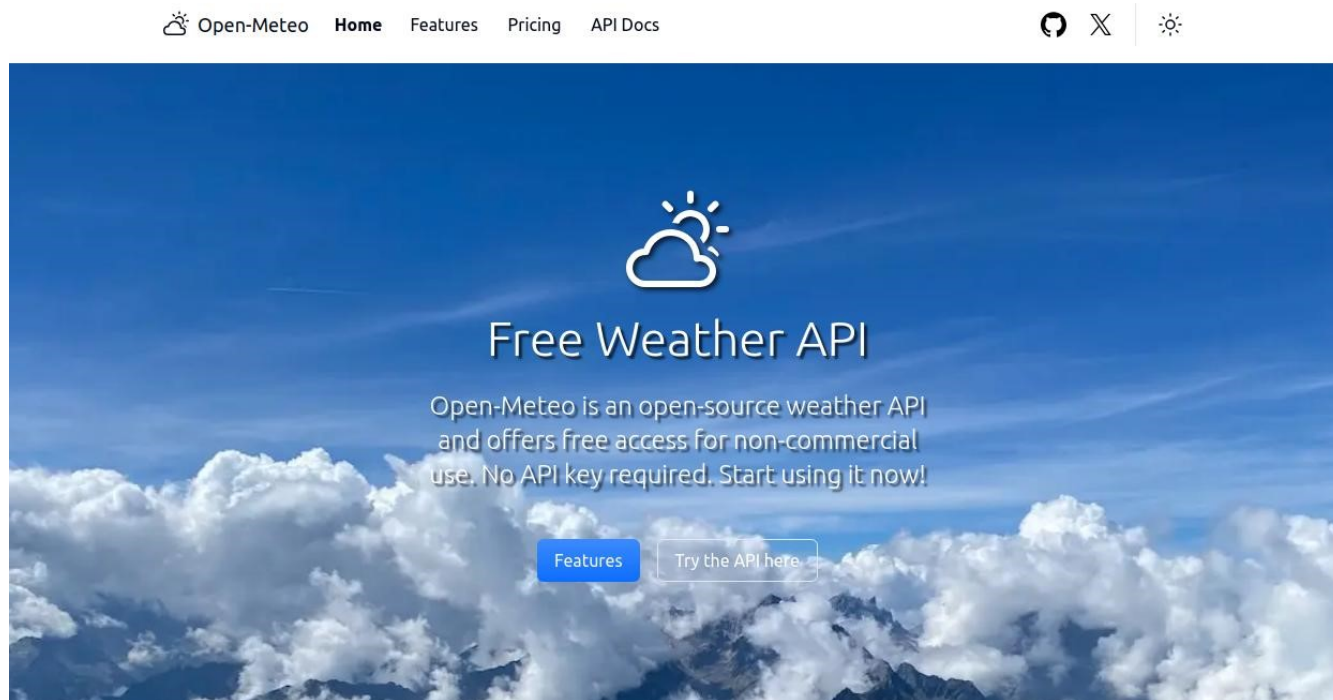
Take JSON/YAML description as input

APL code as output

Use HttpClient internally

Hide “API Stuff”

Demo 1



Demo 2

OpenAI Developers

Home API ▾ Codex ▾ ChatGPT ▾ Resources ▾

API Dashboard ↗ ⚙️ 🔍

Get started

Overview

Quickstart

Models

Pricing

Libraries

Latest: GPT-5.4

Prompt guidance

Core concepts

Text generation

Code generation

Images and vision

Audio and speech

Structured output

Function calling

Responses API

Using tools

API Platform


Developer quickstart

Make your first API request in minutes. Learn the basics of the OpenAI platform.

Get started

javascript ↕

```
1 import OpenAI from "openai";
2 const client = new OpenAI();
3
4 const response = await client.responses.create({
5   model: "gpt-5.4",
6   input: "Write a short bedtime story about a unicorn."
7 });
8
9 console.log(response.output_text);
```



Models

Start with gpt-5.4 for complex reasoning and coding, or choose gpt-5.4-mini and gpt-5.4-nano for lower-latency, lower-cost workloads.

View all

What Else?

OpenAPI REST Servers

Write spec first

Force to think about *design* of API

Tool ecosystem available

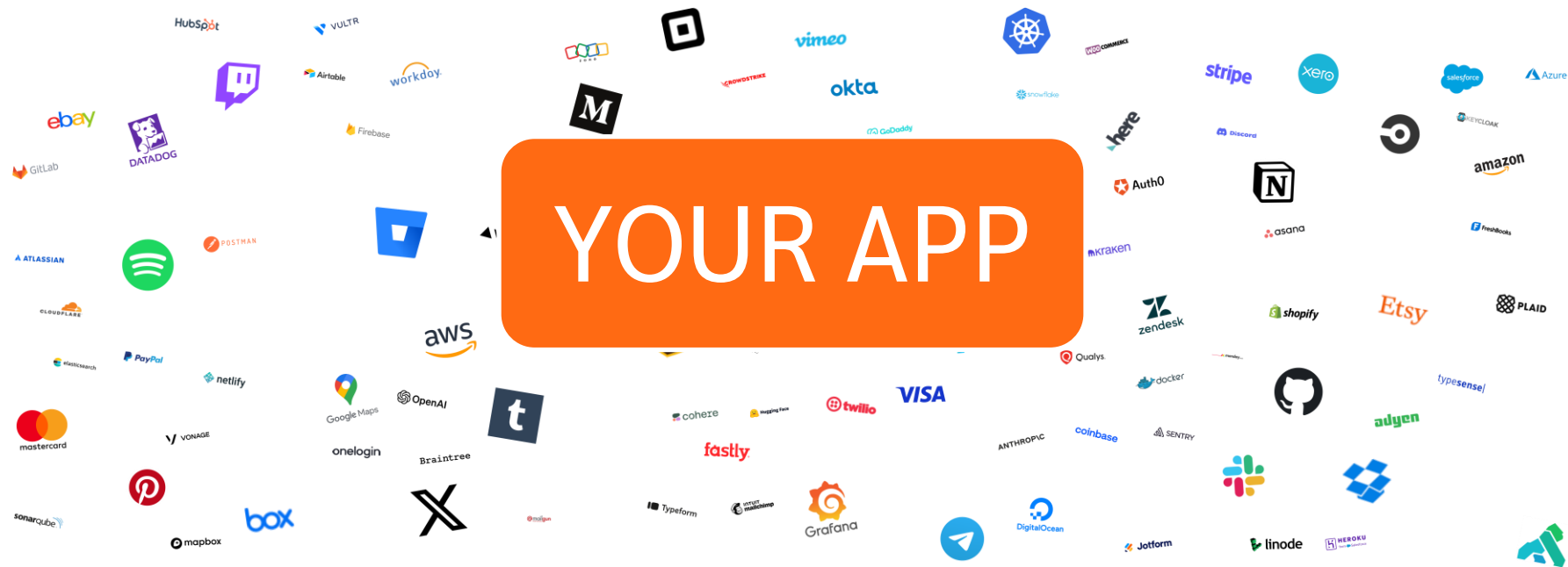
Generation

Documentation

Testing

LLMs work well with OpenAPI

Why should I care?



Why should I care?



APL OpenAPI Services

Stark

Experimental Project

Jarvis

“Register Routes”

Autogenerates OpenAPI JSON

What about APL Servers?

```
▽ result←Get request;segments;root;id
segments←'/'(≠⊢)request.Endpoint
root←>segments
:Select root
:Case 'items'
    :If 1≠segments
        result←ListItems request
    :Else
        id←2>segments
        result←GetItem id request
    :EndIf
:Case 'health'
    result←Health request
:EndSelect
▽
```

```
router←Stark.New ()
routes←[
    'GET' '/health'      'Health'
    'GET' '/items'       'ListItems'
    'GET' '/items/{id}'  'GetItem'
]
routes←router.Register routes

router.Start 8080|
```

What about APL Servers?

```
▽ result←Get request;segments;root;id
segments←'/'(≠⊔)request.Endpoint
root←>segments
:Select root
:Case 'items'
    :If 1=≠segments
        result←ListItems request
    :Else
        id←2>segments
        result←GetItem id request
    :EndIf
:Case 'health'
    result←Health request
:EndSelect
▽
```

```
router←Stark.New ()
routes←[
    GET' /health'      'Health'
    GET' /items'       'ListItems'
    GET' /items/{id}'  'GetItem'
]
routes←router.Register routes

router.Start 8080|
```

What about APL Servers?

```
▽ result←Get request;segments;root;id
segments←'/'(≠⊔)request.Endpoint
root←>segments
:Select root
:Case 'items'
    :If 1≠segments
        result←ListItems request
    :Else
        id←2>segments
        result←GetItem id request
    :EndIf
:Case 'health'
    result←Health request
:EndSelect
▽
```

```
router←Stark.New ()
routes←[
    'GET' '/health' 'Health'
    'GET' '/items' 'ListItems'
    'GET' '/items/{id}' 'GetItem'
]
routes←router.Register routes

router.Start 8080
```

What about APL Servers?

```
▽ result←Get request;segments;root;id
segments←'/'(≠⊔)request.Endpoint
root←>segments
:Select root
:Case 'items'
    :If 1≠segments
        result←ListItems request
    :Else
        id←2>segments
        result←GetItem id request
    :EndIf
:Case 'health'
    result←Health request
:EndSelect
▽
```

```
router←Stark.New ()
routes←[
    'GET' '/health'      'Health'
    'GET' '/items'       'ListItems'
    'GET' '/items/{id}'  'GetItem'
]
routes←router.Register routes

router.Start 8080
```

What about APL Servers?

```
CreateItemSchema←(  
  summary: 'Create a new item'  
  tags: ('items'◇)  
  body: (type: 'object' ◇ required: 'name' 'price'  
    properties: (  
      name: (type: 'string')  
      price: (type: 'number')  
    )  
  )  
  responses: (  
    Δ201: (type: 'object'  
      properties: (  
        id: (type: 'integer')  
        name: (type: 'string')  
        price: (type: 'number')  
        tags: (type: 'array' ◇ items: (type: 'string'))  
      )  
    )  
  )  
)
```

Example Items API 1.0.0 OAS 3.0

status

invoices

GET /customer/{cust_id}/invoice/{inv_id} Get a customer invoice

items

GET /items List all items

POST /items Create a new item

DELETE /items/{id} Delete an item

GET /items/{id} Get an item by ID

PUT /items/{id} Update an item

Future?