DYALOG
Eastbourne 2026

Making Web Services More Accessible



Holden Hoover

holden@dyalog.com

The Plan

- Web Services Overview
- Consuming
- Serving

About Me

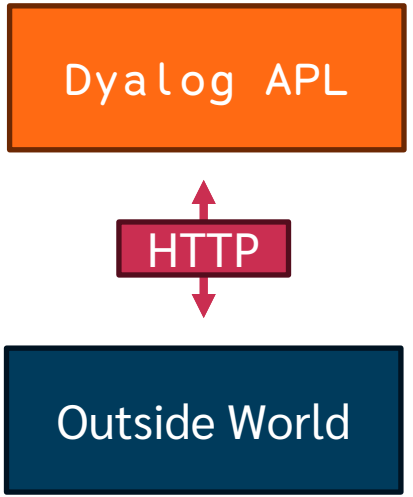


- 2024 APL Forge Winner
- UWaterloo Computer Engineering
- Development Intern @ Dyalog Ltd
  - Tools for Web Services
  - Working on Link
  - And other stuff...



Web Services

- Simple concept: APL talks to the world over HTTP
- You call out
  - fetch data, run a model, make a payment
- The world calls in
  - someone runs your APL logic
- HTTP doesn't care what's on either end



REST

Representational State Transfer (REST) architectural style

REST

Representational State Transfer (REST) architectural style

- A **convention** for using HTTP, not a standard or a library
- URL is the **noun**, method is the **verb**
  - GET / POST / PUT / DELETE
- You get back a **representation**
  - JSON, XML, CSV, an image

```
GET  /v1/forecast
verb noun
?latitude=50.77
&longitude=0.28
&current=temperature_2m
```

stateless

```
200 OK
{"temperature_2m":14.3}
representation (JSON)
```

The Plan

- Web Services Overview
- **Consuming**
- Serving

Consuming

Why should *I* care?

Why should I care?



Stripe: payments









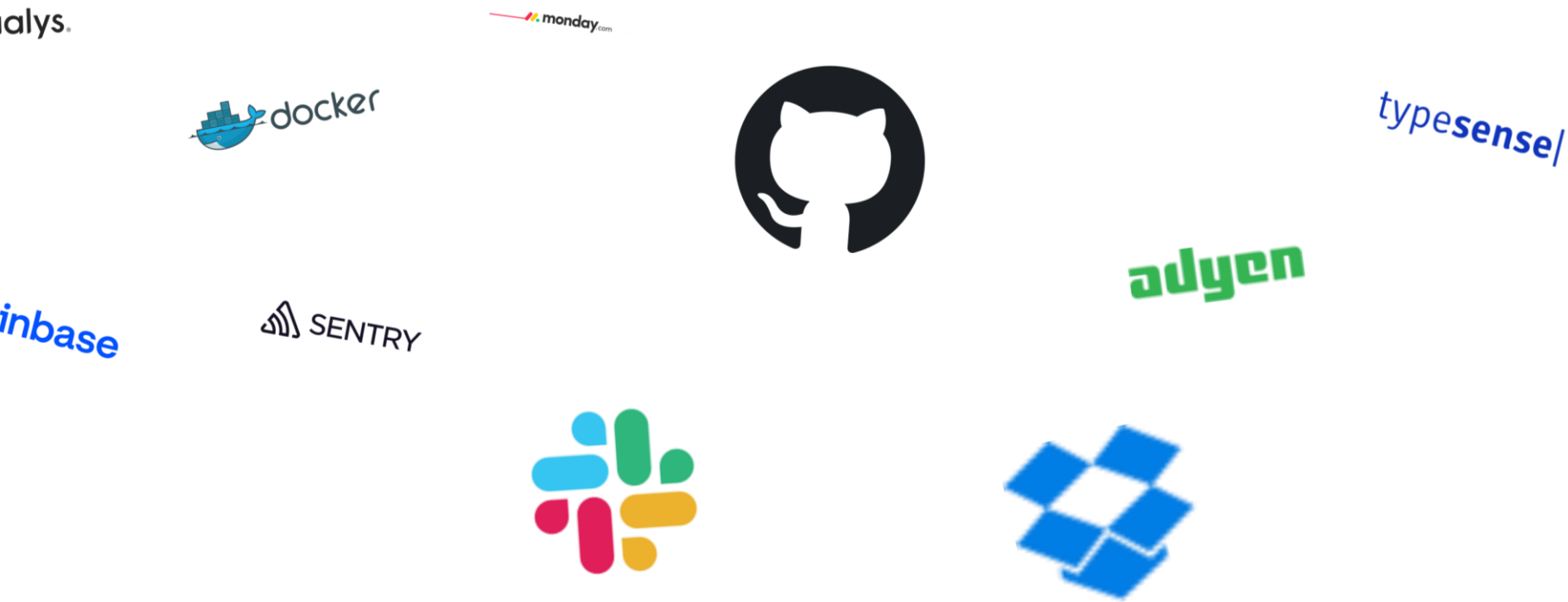








GitHub: code hosting



Twilio: messaging & voice





















OpenAI: AI models





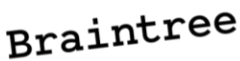














Why should I care?



HttpCommand

```
r←HttpCommand.Get url
```

Wouldn’t it be nice?

Introducing: OpenAPI*

- A contract: “it will act like this”
- Readable by machine ^ human
- Clients generated from the spec
- Spec changes? Regenerate the client

** Formerly known as Swagger*

What's in the contract?

One operation from an OpenAPI file, written in YAML or JSON

```yaml
paths:
  /users/{userId}:
    get:
      operationId: getUser
      summary: Get a user by ID
      parameters:
        - name: userId
          in: path
          required: true
          schema: {type: integer}
      responses:
        '200':
          description: The user
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/User'
```

Operation path + method

Docs summary

Inputs parameters

Outputs responses

What's in the contract?

Operation the path, the HTTP method and a name

```yaml
paths:
  /users/{userId}:
    get:
      operationId: getUser
      summary: Get a user by ID
      parameters:
        - name: userId
          in: path
```

What's in the contract?

Docs human-readable text beside the machine-readable parts

```
get:
  operationId: getUser
  summary: Get a user by ID
  parameters:
    - name: userId
      in: path
      required: true
      schema: {type: integer}
```

What's in the contract?

Inputs each parameter, where it goes and its type

```
operationId: getUser
summary: Get a user by ID
parameters:
  - name: userId
    in: path
    required: true
    schema: {type: integer}
responses:
```

What's in the contract?

Outputs what comes back for each response code, and its shape

```
        schema: {type: integer}
    responses:
      '200':
        description: The user
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/User'
```

What's in the contract?

× every endpoint = the whole API, described in one file

```
GET     /users
POST    /users
GET     /users/{userId}   ← the one we just read
PUT     /users/{userId}
DELETE  /users/{userId}
GET     /users/{userId}/orders
POST    /orders
```

Who uses it?

Who uses it?

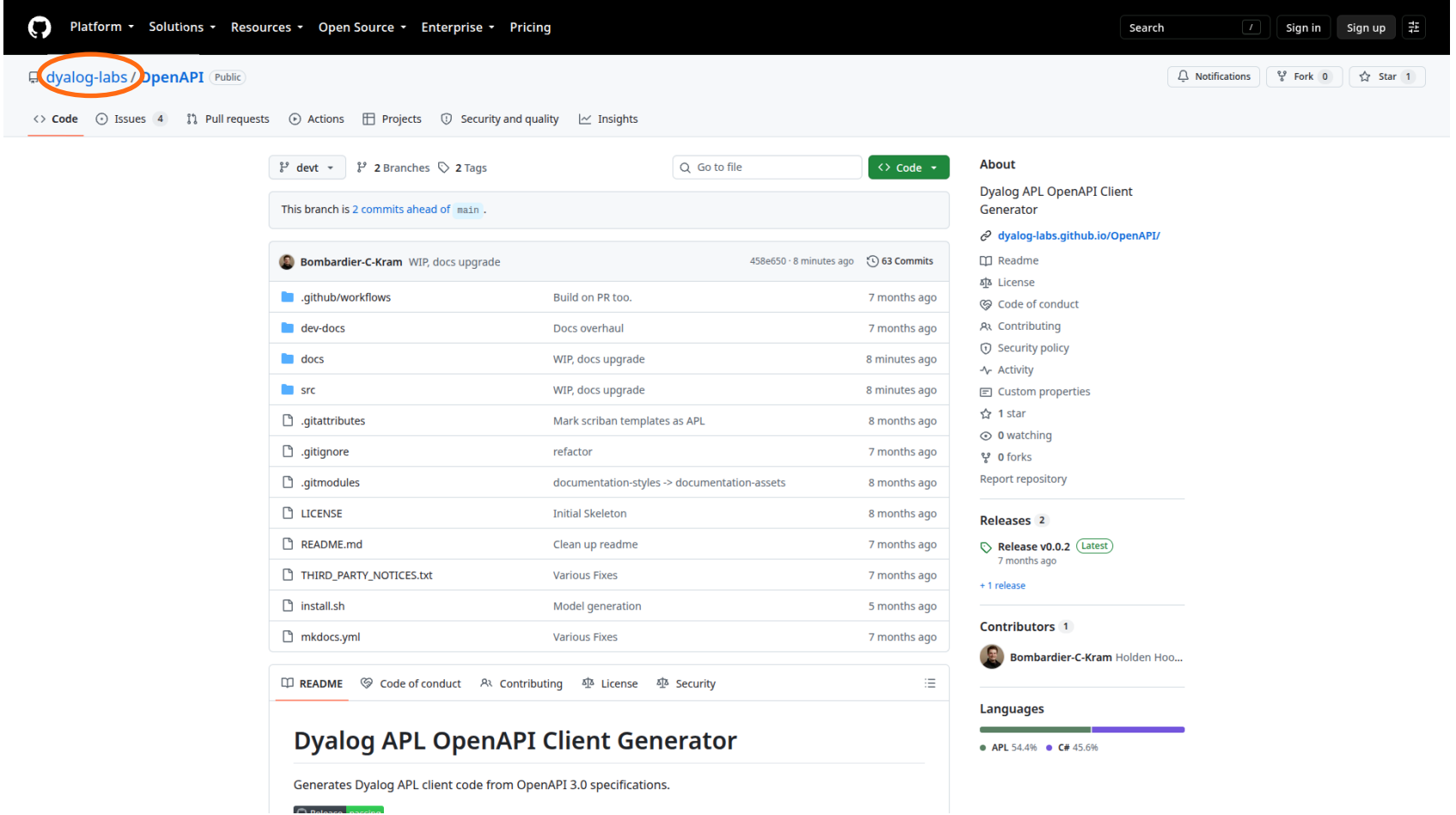

HubSpot
VULTR
vimeo
stripe
xero
salesforce
Azure
ebay
DATADOG
Airtable
workday.
okta
Firebase
GitLab
FedEx
GoDaddy
here
Discord
KEYCLOAK
amazon
Google Cloud
WISE
Auth0
Trello
asana
FreshBooks
ATLASSIAN
POSTMAN
Vercel
HETZNER
reddit
KRAKEN
CLOUDFLARE
aws
zendesk
shopify
Etsy
PLAID
PayPal
netlify
imgur
GIPHY
Qualys.
docker
typesense
Google Maps
OpenAI
twilio
VISA
cohere
mastercard
VONAGE
onelogin
Braintree
fastly
ANTHROPIC
coinbase
SENTRY
adyen
Typeform
Grafana
DigitalOcean
Jotform
linode
HEROKU
sonarqube
box
mapbox

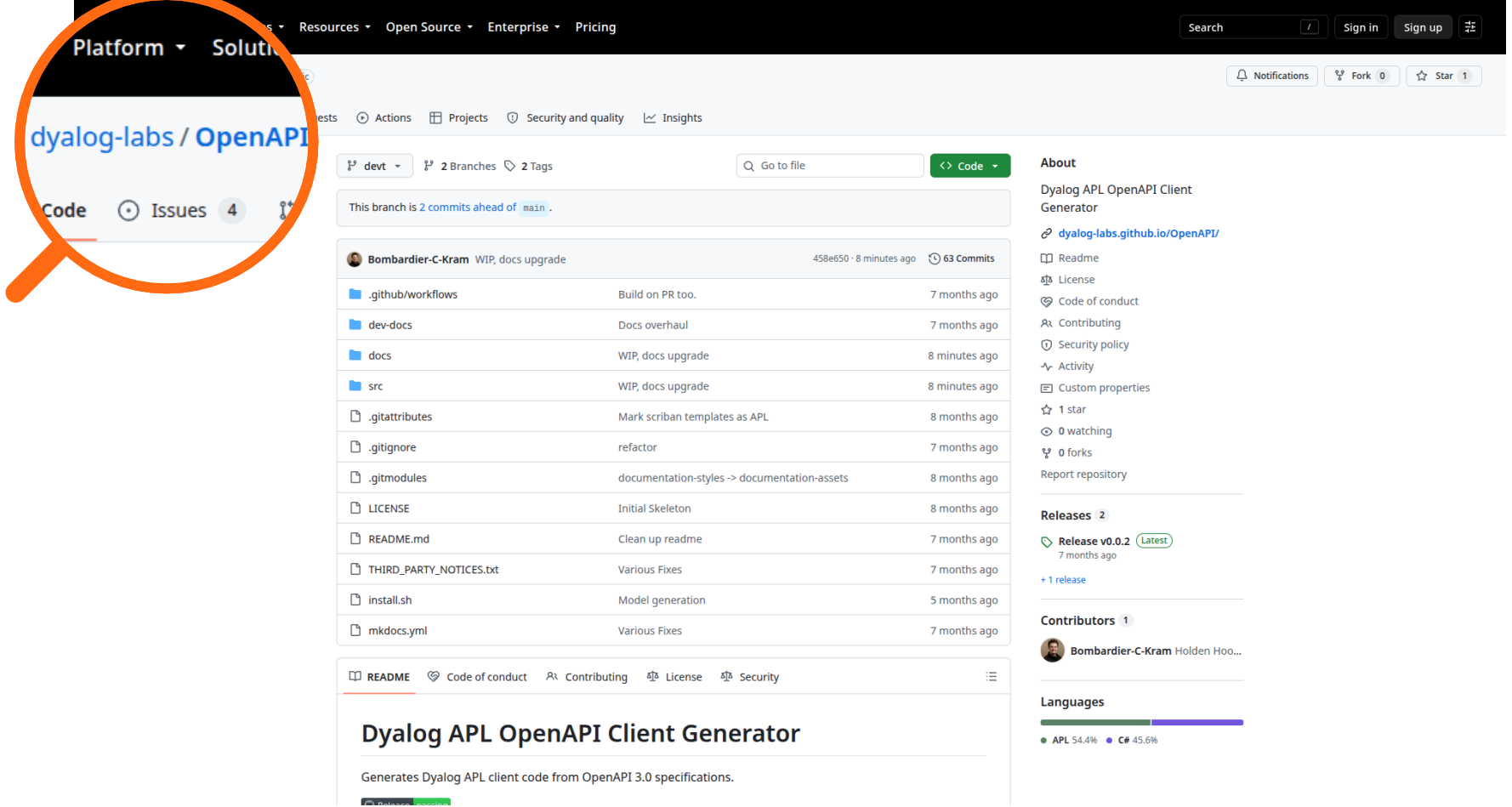


+


OPENAPI
INITIATIVE

?


Platform Solutions Resources Open Source Enterprise Pricing
Search / Sign in Sign up
dyalog-labs / OpenAPI Public
Notifications Fork 0 Star 1
Code Issues 4 Pull requests Actions Projects Security and quality Insights
devt 2 Branches 2 Tags Go to file Code
This branch is 2 commits ahead of main.
Bombardier-C-Kram WIP, docs upgrade 458e650 · 8 minutes ago 63 Commits
.github/workflows Build on PR too. 7 months ago
dev-docs Docs overhaul 7 months ago
docs WIP, docs upgrade 8 minutes ago
src WIP, docs upgrade 8 minutes ago
.gitattributes Mark scriban templates as APL 8 months ago
.gitignore refactor 7 months ago
.gitmodules documentation-styles -> documentation-assets 8 months ago
LICENSE Initial Skeleton 8 months ago
README.md Clean up readme 7 months ago
THIRD_PARTY_NOTICES.txt Various Fixes 7 months ago
install.sh Model generation 5 months ago
mkdocs.yml Various Fixes 7 months ago
README Code of conduct Contributing License Security
Dyalog APL OpenAPI Client Generator
Generates Dyalog APL client code from OpenAPI 3.0 specifications.
About
Dyalog APL OpenAPI Client Generator
dyalog-labs.github.io/OpenAPI/
Readme
License
Code of conduct
Contributing
Security policy
Activity
Custom properties
1 star
0 watching
0 forks
Report repository
Releases 2
Release v0.0.2 Latest
7 months ago
+ 1 release
Contributors 1
Bombardier-C-Kram Holden Hoo...
Languages
APL 54.4% C# 45.6%


Platform
Resources
Open Source
Enterprise
Pricing
Search
Sign in
Sign up
dyalog-labs / OpenAPI
Notifications
Fork 0
Star 1
Code
Issues 4
Actions
Projects
Security and quality
Insights
devt
2 Branches
2 Tags
Go to file
Code
This branch is 2 commits ahead of main.
Bombardier-C-Kram WIP, docs upgrade
458e650 · 8 minutes ago
63 Commits
.github/workflows Build on PR too. 7 months ago
dev-docs Docs overhaul 7 months ago
docs WIP, docs upgrade 8 minutes ago
src WIP, docs upgrade 8 minutes ago
.gitattributes Mark scriban templates as APL 8 months ago
.gitignore refactor 7 months ago
.gitmodules documentation-styles -> documentation-assets 8 months ago
LICENSE Initial Skeleton 8 months ago
README.md Clean up readme 7 months ago
THIRD_PARTY_NOTICES.txt Various Fixes 7 months ago
install.sh Model generation 5 months ago
mkdocs.yml Various Fixes 7 months ago
README
Code of conduct
Contributing
License
Security
Dyalog APL OpenAPI Client Generator
Generates Dyalog APL client code from OpenAPI 3.0 specifications.
About
Dyalog APL OpenAPI Client Generator
dyalog-labs.github.io/OpenAPI/
Readme
License
Code of conduct
Contributing
Security policy
Activity
Custom properties
1 star
0 watching
0 forks
Report repository
Releases 2
Release v0.0.2 Latest
7 months ago
+ 1 release
Contributors 1
Bombardier-C-Kram Holden Hoo...
Languages
APL 54.4%
C# 45.6%

APL + OpenAPI = an APL client

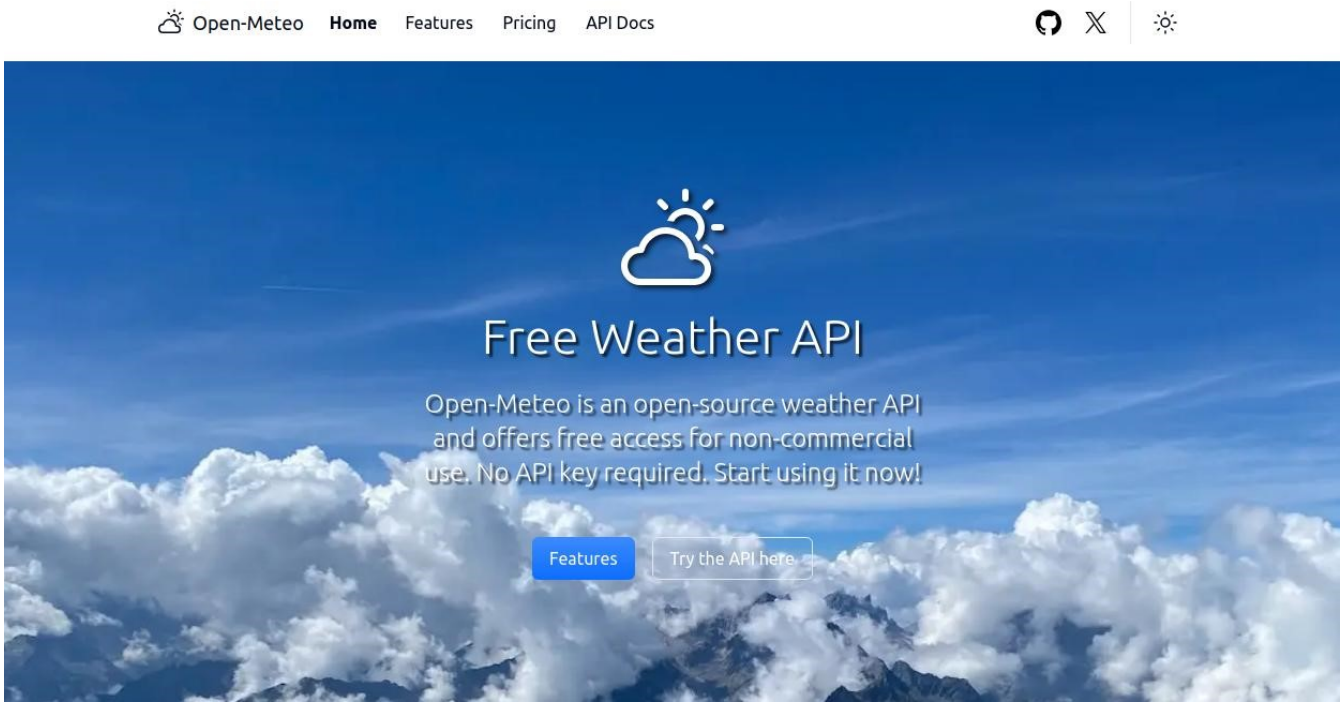
OpenAPI spec

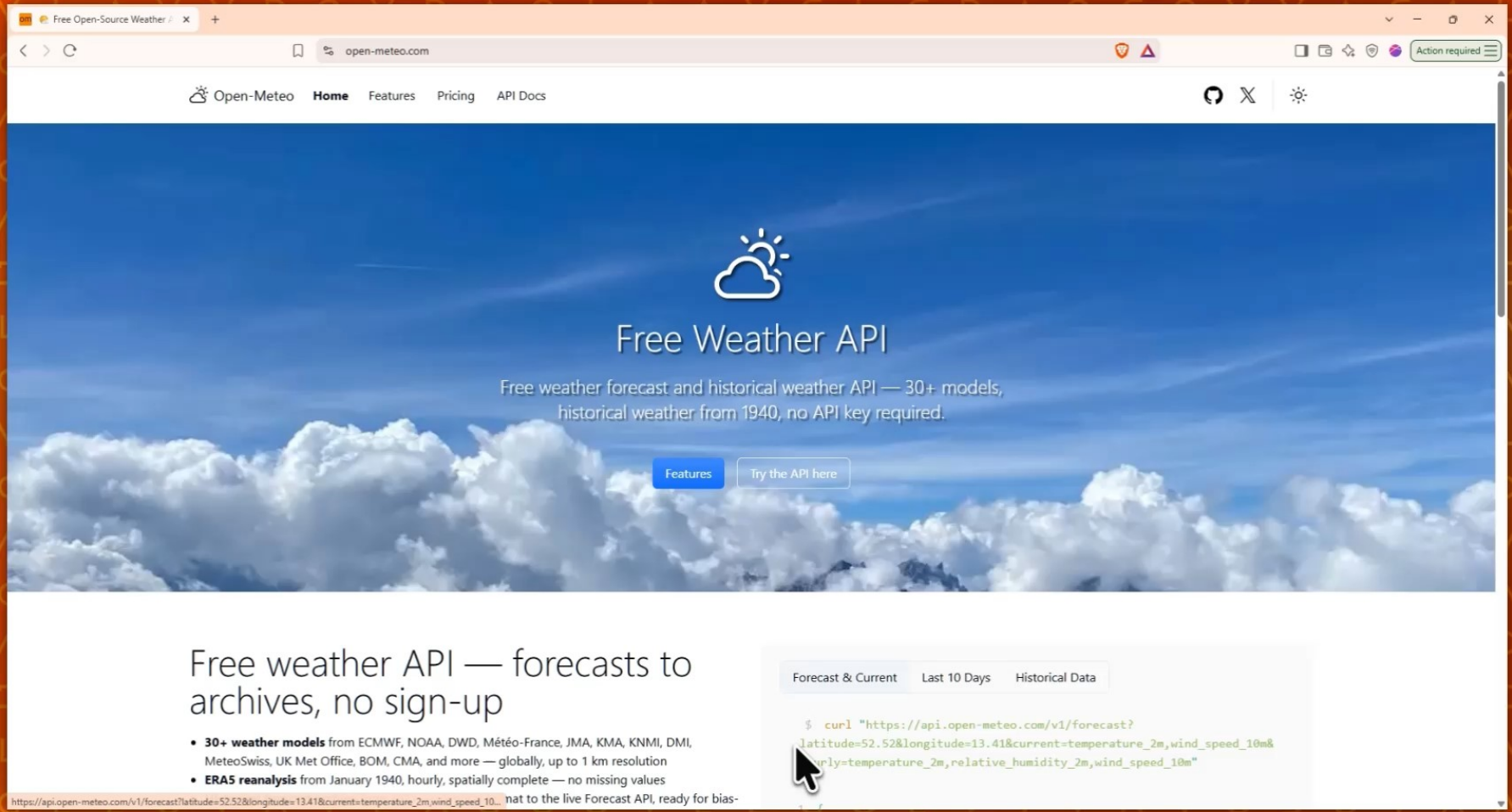
```
/users/{userId}:
  get:
    tags: [users]
    operationId: getUser
```

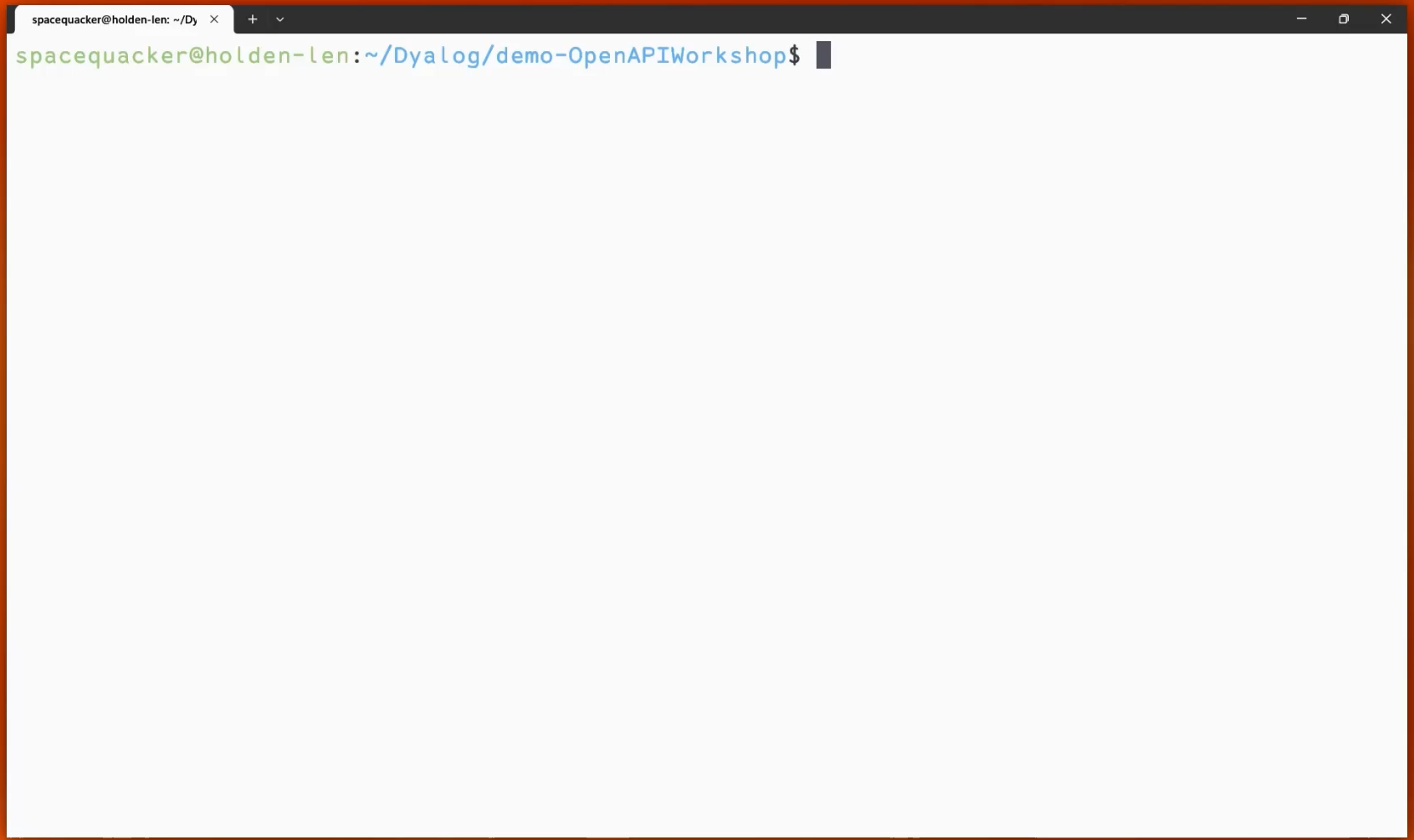
Generated APL client, in the session

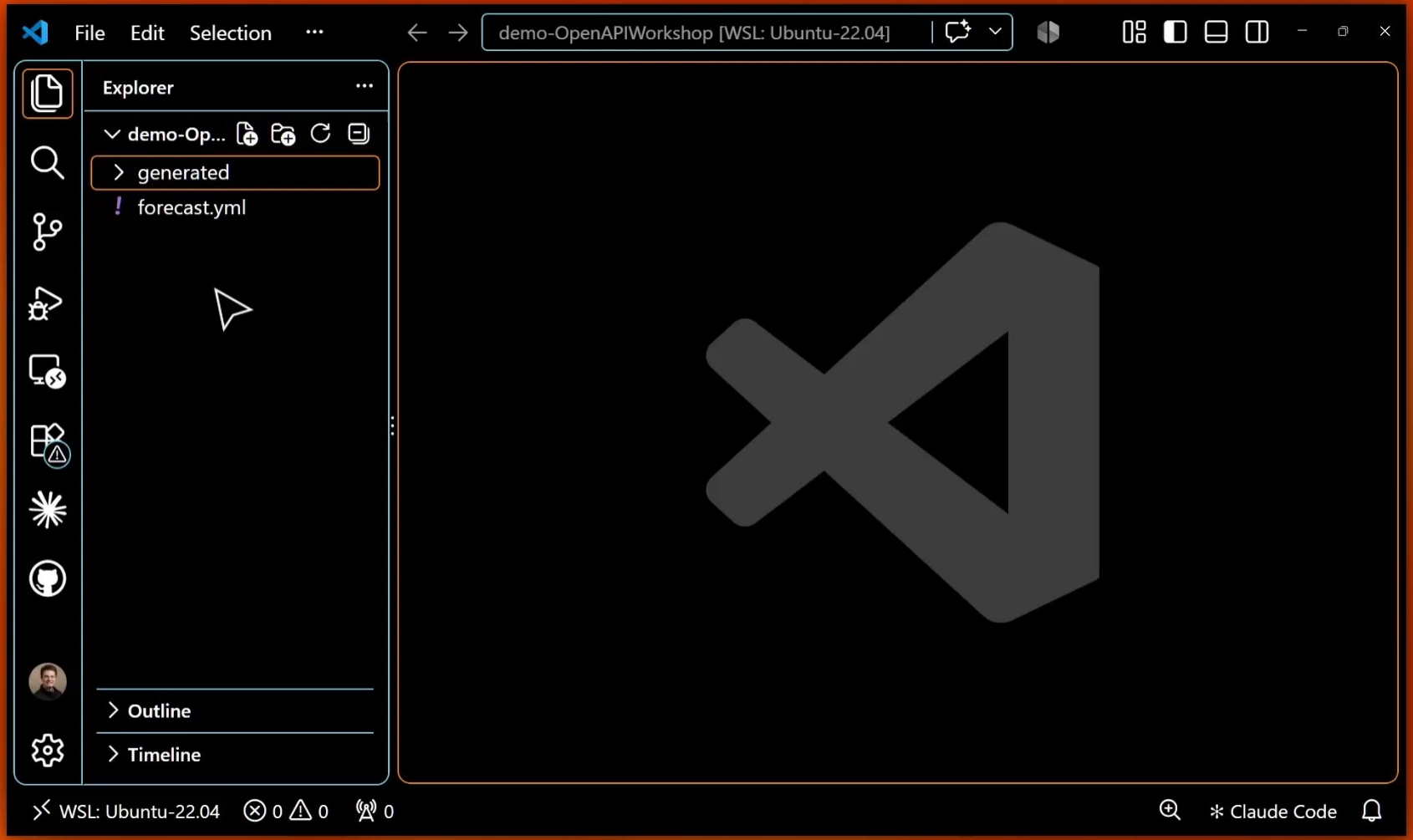
```
      config←(baseUrl:'https://api.example.com')
      api←Example.Client.New config
      r←api.users.getUser (userId:42)
      r.Data.name
Mort B. Monkerger
```

Example 1


Open-Meteo
Home
Features
Pricing
API Docs
Free Weather API
Open-Meteo is an open-source weather API and offers free access for non-commercial use. No API key required. Start using it now!
Features
Try the API here


Free Open-Source Weather
open-meteo.com
Action required
Open-Meteo
Home
Features
Pricing
API Docs
Free Weather API
Free weather forecast and historical weather API — 30+ models, historical weather from 1940, no API key required.
Features
Try the API here
Free weather API — forecasts to archives, no sign-up
30+ weather models from ECMWF, NOAA, DWD, Météo-France, JMA, KMA, KNMI, DMI, MeteoSwiss, UK Met Office, BOM, CMA, and more — globally, up to 1 km resolution
ERA5 reanalysis from January 1940, hourly, spatially complete — no missing values
Forecast & Current
Last 10 Days
Historical Data
$ curl "https://api.open-meteo.com/v1/forecast?latitude=52.52&longitude=13.41&current=temperature_2m,wind_speed_10m&hourly=temperature_2m,relative_humidity_2m,wind_speed_10m"

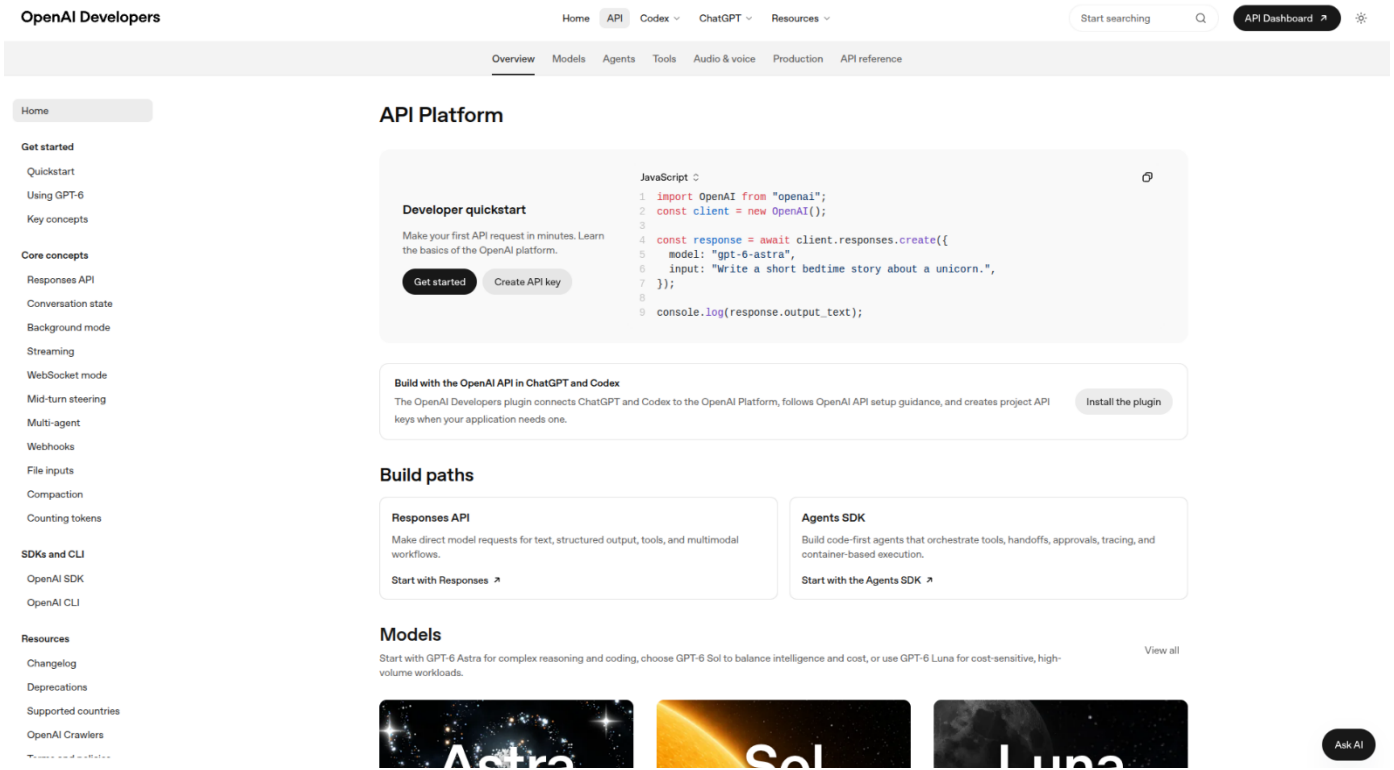

spacequacker@holden-len: ~/Dy
spacequacker@holden-len:~/Dyalog/demo-OpenAPIWorkshop$


File
Edit
Selection
demo-OpenAPIWorkshop [WSL: Ubuntu-22.04]
Explorer
demo-Op...
generated
forecast.yml
Outline
Timeline
WSL: Ubuntu-22.04
0
0
0
Claude Code

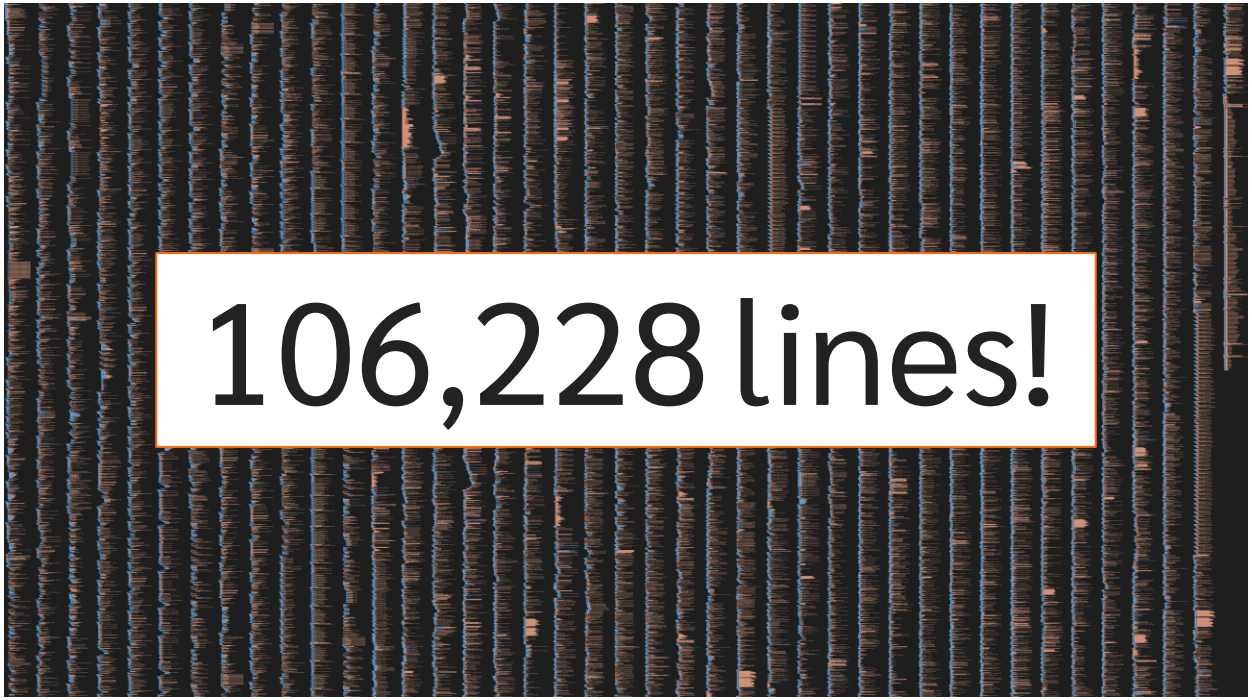
Using the OpenMeteo client

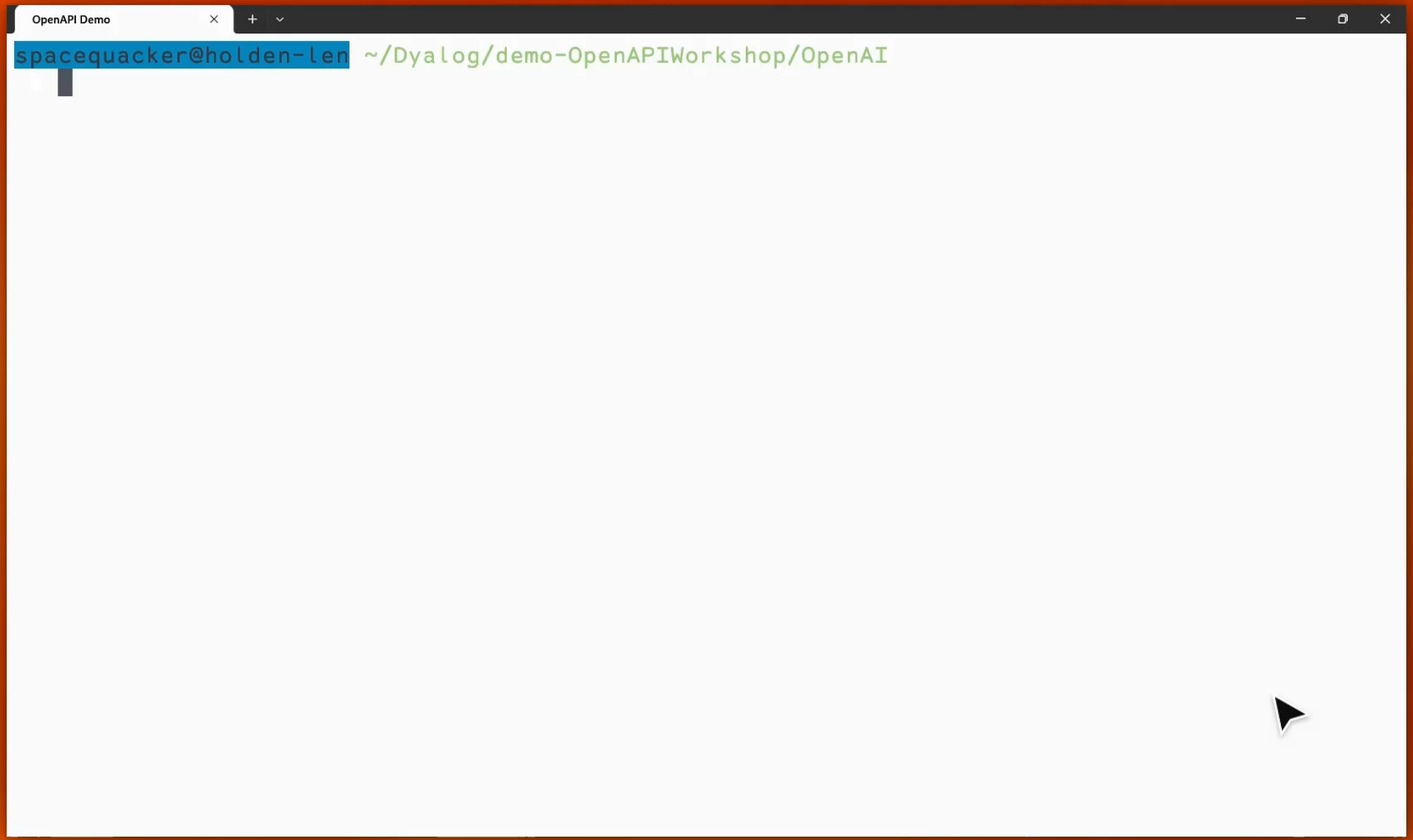
```
      config←(baseUrl:'https://api.open-meteo.com')
      meteo←OpenMeteo.Client.New config
      args←(latitude:'50.76' ⋄ longitude:'0.29')
      args.current←'temperature_2m'
      r←meteo.weatherForecastAPIs.GetV1Forecast args
      r
[rc: 0 | msg:  | HTTP Status: 200 "OK" | ≠Data: 1 (namespace)]
      r.Data.⎕NL -⍳9
 current  current_units  elevation  generationtime_ms  latitude
 longitude  timezone  timezone_abbreviation  utc_offset_seconds
      r.Data.(current.temperature_2m current_units.temperature_2m)
11.6  °C
```

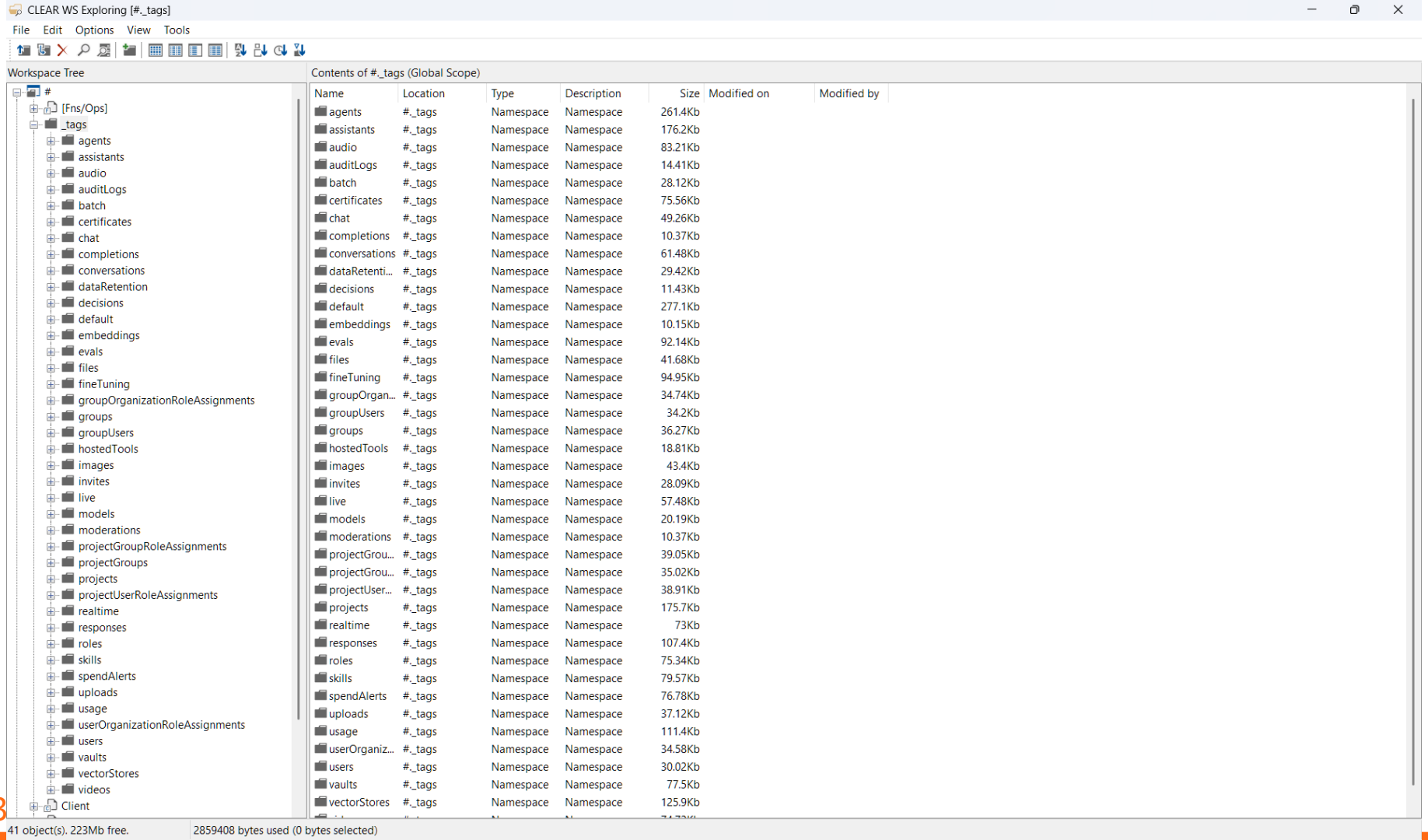
Example 2


OpenAI Developers
Home
API
Codex
ChatGPT
Resources
Start searching
API Dashboard
Overview
Models
Agents
Tools
Audio & voice
Production
API reference
Home
Get started
Quickstart
Using GPT-6
Key concepts
Core concepts
Responses API
Conversation state
Background mode
Streaming
WebSocket mode
Mid-turn steering
Multi-agent
Webhooks
File inputs
Compaction
Counting tokens
SDKs and CLI
OpenAI SDK
OpenAI CLI
Resources
Changelog
Deprecations
Supported countries
OpenAI Crawlers
API Platform
Developer quickstart
Make your first API request in minutes. Learn the basics of the OpenAI platform.
Get started
Create API key
JavaScript
import OpenAI from "openai";
const client = new OpenAI();
const response = await client.responses.create({
model: "gpt-6-astra",
input: "Write a short bedtime story about a unicorn.",
});
console.log(response.output_text);
Build with the OpenAI API in ChatGPT and Codex
The OpenAI Developers plugin connects ChatGPT and Codex to the OpenAI Platform, follows OpenAI API setup guidance, and creates project API keys when your application needs one.
Install the plugin
Build paths
Responses API
Make direct model requests for text, structured output, tools, and multimodal workflows.
Start with Responses
Agents SDK
Build code-first agents that orchestrate tools, handoffs, approvals, tracing, and container-based execution.
Start with the Agents SDK
Models
Start with GPT-6 Astra for complex reasoning and coding, choose GPT-6 Sol to balance intelligence and cost, or use GPT-6 Luna for cost-sensitive, high-volume workloads.
View all
Astra
Sol
Luna
Ask AI

OpenAI OpenAPI Spec




OpenAPI Demo
spacequacker@holden-len ~/Dyalog/demo-OpenAPIWorkshop/OpenAI


CLEAR WS Exploring [#._tags]
File Edit Options View Tools
Workspace Tree
Contents of #._tags (Global Scope)
[Fns/Ops] _tags agents assistants audio auditLogs batch certificates chat completions conversations dataRetention decisions default embeddings evals files fineTuning groupOrganizationRoleAssignments groups groupUsers hostedTools images invites live models moderations projectGroupRoleAssignments projectGroups projects projectUserRoleAssignments realtime responses roles skills spendAlerts uploads usage userOrganizationRoleAssignments users vaults vectorStores videos Client
Name Location Type Description Size Modified on Modified by
agents #._tags Namespace Namespace 261.4Kb
assistants #._tags Namespace Namespace 176.2Kb
audio #._tags Namespace Namespace 83.21Kb
auditLogs #._tags Namespace Namespace 14.41Kb
batch #._tags Namespace Namespace 28.12Kb
certificates #._tags Namespace Namespace 75.56Kb
chat #._tags Namespace Namespace 49.26Kb
completions #._tags Namespace Namespace 10.37Kb
conversations #._tags Namespace Namespace 61.48Kb
dataRetenti... #._tags Namespace Namespace 29.42Kb
decisions #._tags Namespace Namespace 11.43Kb
default #._tags Namespace Namespace 277.1Kb
embeddings #._tags Namespace Namespace 10.15Kb
evals #._tags Namespace Namespace 92.14Kb
files #._tags Namespace Namespace 41.68Kb
fineTuning #._tags Namespace Namespace 94.95Kb
groupOrgan... #._tags Namespace Namespace 34.74Kb
groupUsers #._tags Namespace Namespace 34.2Kb
groups #._tags Namespace Namespace 36.27Kb
hostedTools #._tags Namespace Namespace 18.81Kb
images #._tags Namespace Namespace 43.4Kb
invites #._tags Namespace Namespace 28.09Kb
live #._tags Namespace Namespace 57.48Kb
models #._tags Namespace Namespace 20.19Kb
moderations #._tags Namespace Namespace 10.37Kb
projectGrou... #._tags Namespace Namespace 39.05Kb
projectGrou... #._tags Namespace Namespace 35.02Kb
projectUser... #._tags Namespace Namespace 38.91Kb
projects #._tags Namespace Namespace 175.7Kb
realtime #._tags Namespace Namespace 73Kb
responses #._tags Namespace Namespace 107.4Kb
roles #._tags Namespace Namespace 75.34Kb
skills #._tags Namespace Namespace 79.57Kb
spendAlerts #._tags Namespace Namespace 76.78Kb
uploads #._tags Namespace Namespace 37.12Kb
usage #._tags Namespace Namespace 111.4Kb
userOrganiz... #._tags Namespace Namespace 34.58Kb
users #._tags Namespace Namespace 30.02Kb
vaults #._tags Namespace Namespace 77.5Kb
vectorStores #._tags Namespace Namespace 125.9Kb
41 object(s). 223Mb free.
2859408 bytes used (0 bytes selected)

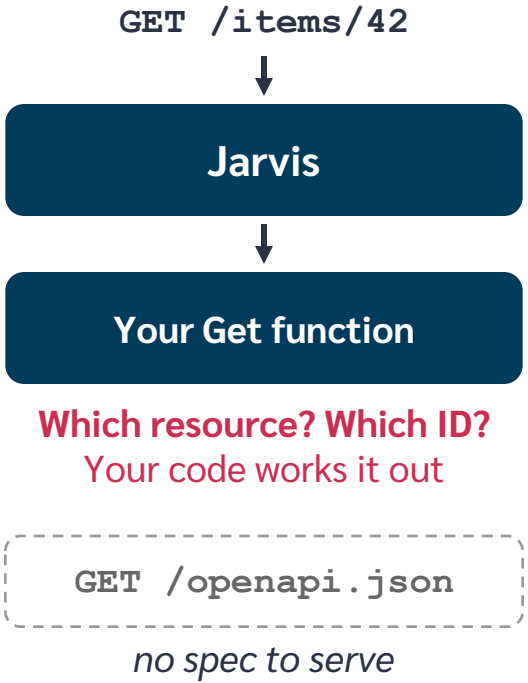
The Plan

- Web Services Overview
- Consuming
- **Serving**

Serving

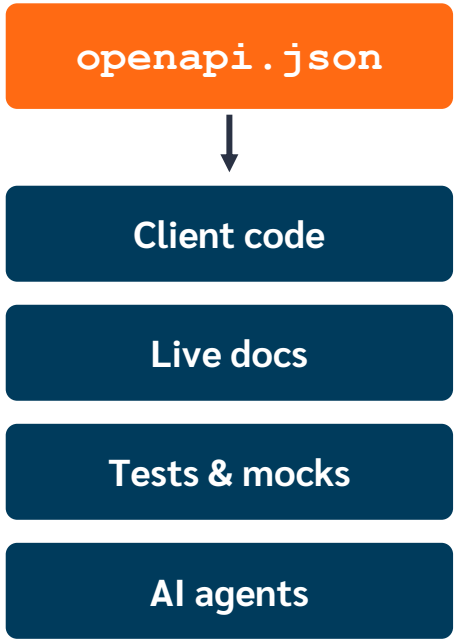
Jarvis

- Dyalog's HTTP server: turns APL functions into a web service
- Sessions, auth, CORS, and more, out of the box
- **JSON mode:** one function per endpoint, named after the URL
- **REST mode:** one function per HTTP method
  - GET /items and GET /items/42 both arrive at the same Get function
  - **Your code has to read the URL to work out what was asked for**



Serving with OpenAPI

- Design first, code second
  - The spec forces you to decide resources, inputs & outputs up front
- Nobody has to learn APL to use it
  - Any consumer, any language can generate a client from your spec
- Documentation that can't go stale
  - Generated from the running service; always true
- AI-ready
  - A spec is exactly what an LLM or coding agent needs to call your service correctly



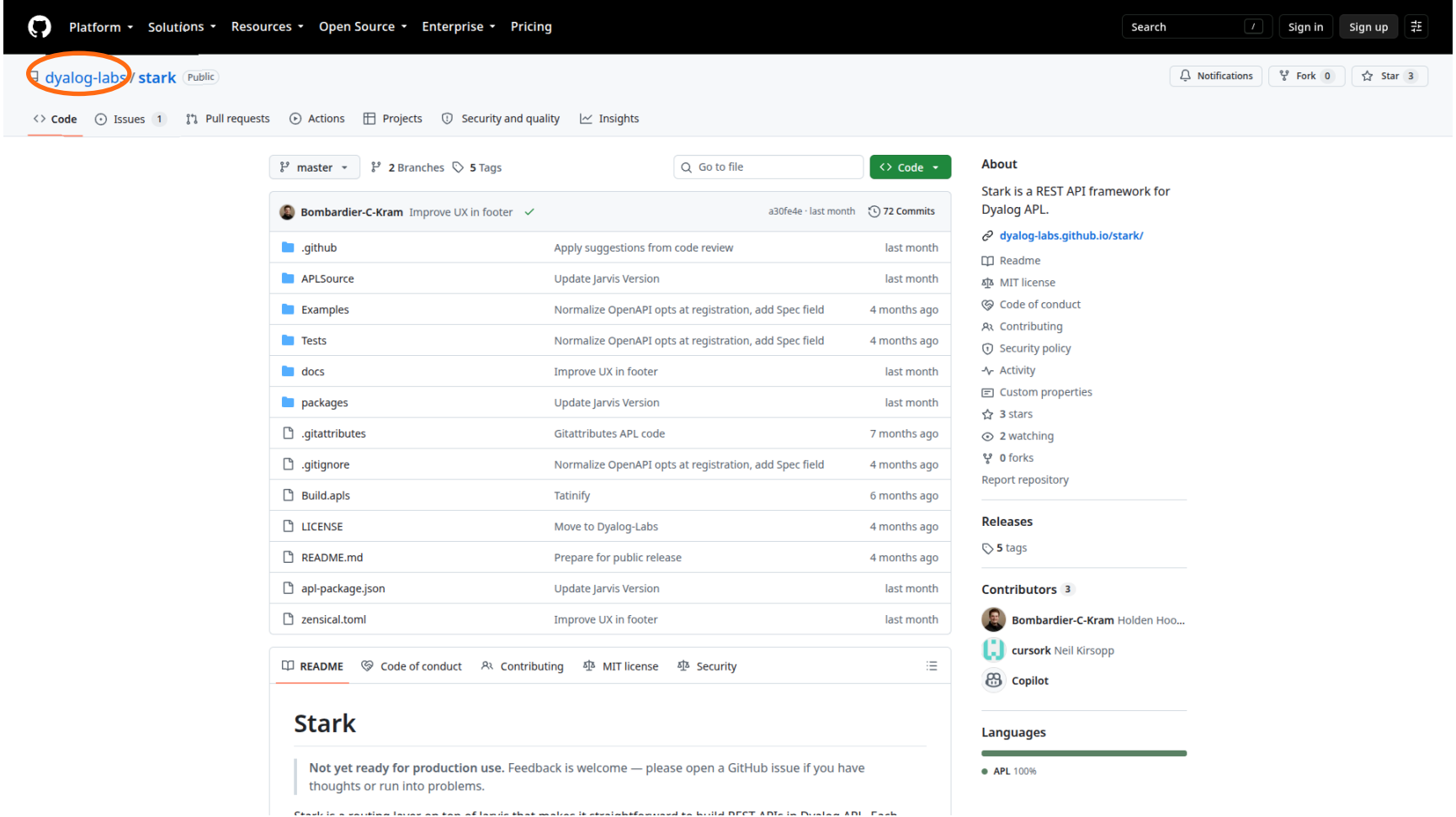
What happens when you combine them?

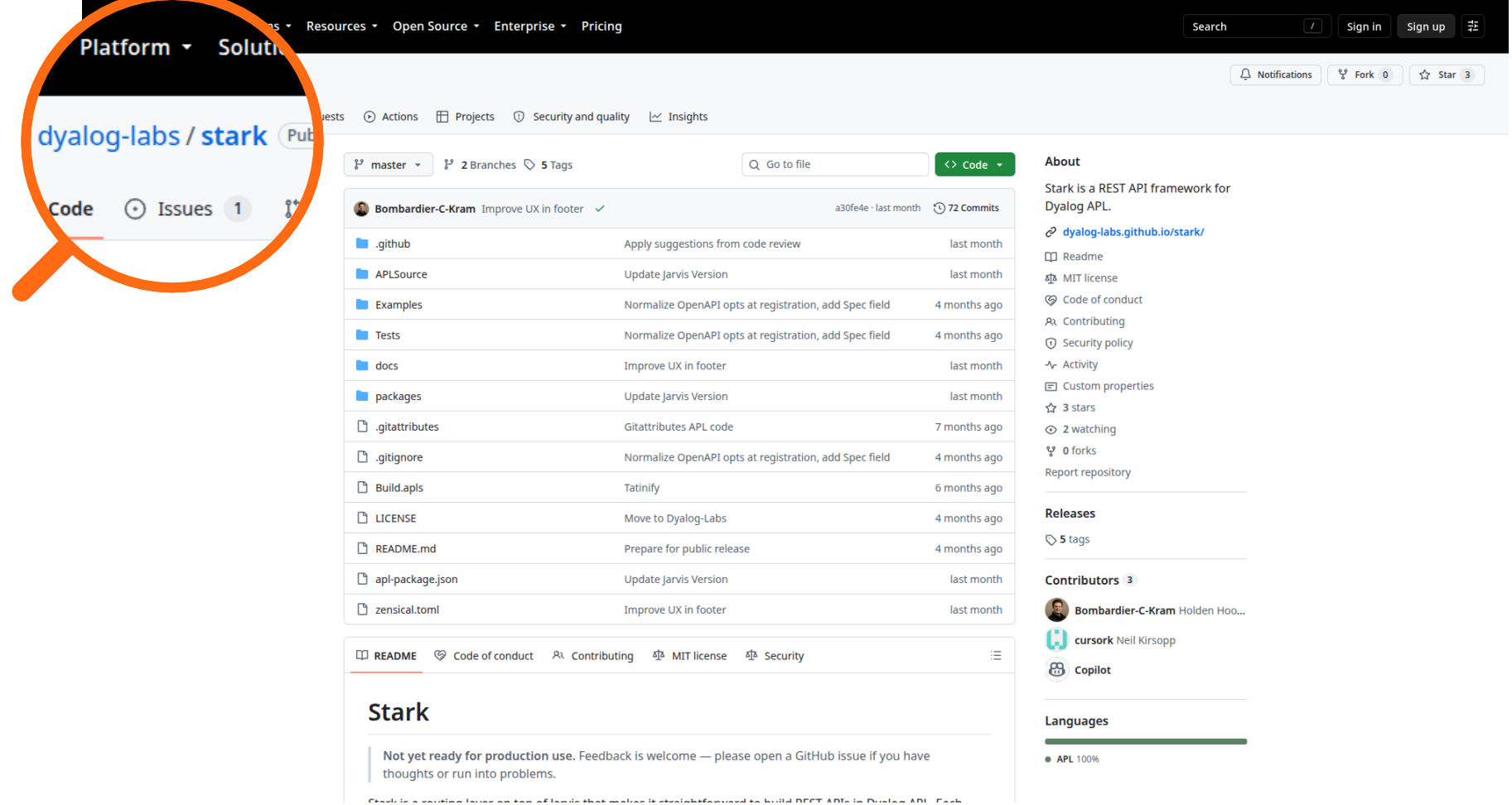


+

?



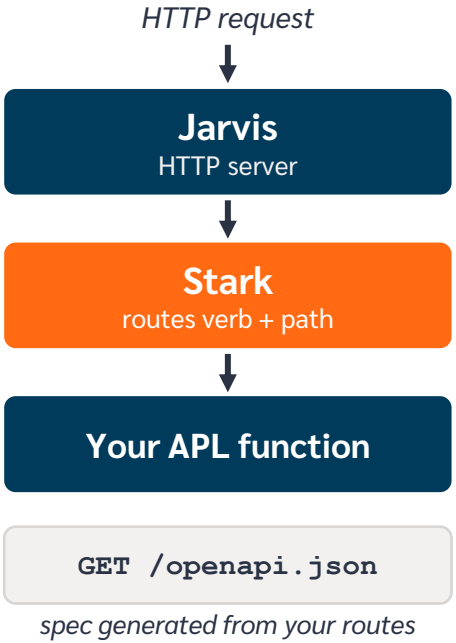

Platform
Solutions
Resources
Open Source
Enterprise
Pricing
Search
Sign in
Sign up
dyalog-labs / stark
Public
Notifications
Fork 0
Star 3
Code
Issues 1
Pull requests
Actions
Projects
Security and quality
Insights
master
2 Branches
5 Tags
Go to file
Code
Bombardier-C-Kram Improve UX in footer
a30fe4e · last month
72 Commits
.github Apply suggestions from code review last month
APLSource Update Jarvis Version last month
Examples Normalize OpenAPI opts at registration, add Spec field 4 months ago
Tests Normalize OpenAPI opts at registration, add Spec field 4 months ago
docs Improve UX in footer last month
packages Update Jarvis Version last month
.gitattributes Gitattributes APL code 7 months ago
.gitignore Normalize OpenAPI opts at registration, add Spec field 4 months ago
Build.apls Tatinify 6 months ago
LICENSE Move to Dyalog-Labs 4 months ago
README.md Prepare for public release 4 months ago
apl-package.json Update Jarvis Version last month
zensical.toml Improve UX in footer last month
README
Code of conduct
Contributing
MIT license
Security
Stark
Not yet ready for production use. Feedback is welcome — please open a GitHub issue if you have thoughts or run into problems.
About
Stark is a REST API framework for Dyalog APL.
dyalog-labs.github.io/stark/
Readme
MIT license
Code of conduct
Contributing
Security policy
Activity
Custom properties
3 stars
2 watching
0 forks
Report repository
Releases
5 tags
Contributors 3
Bombardier-C-Kram Holden Hoo...
cursork Neil Kirsopp
Copilot
Languages
APL 100%



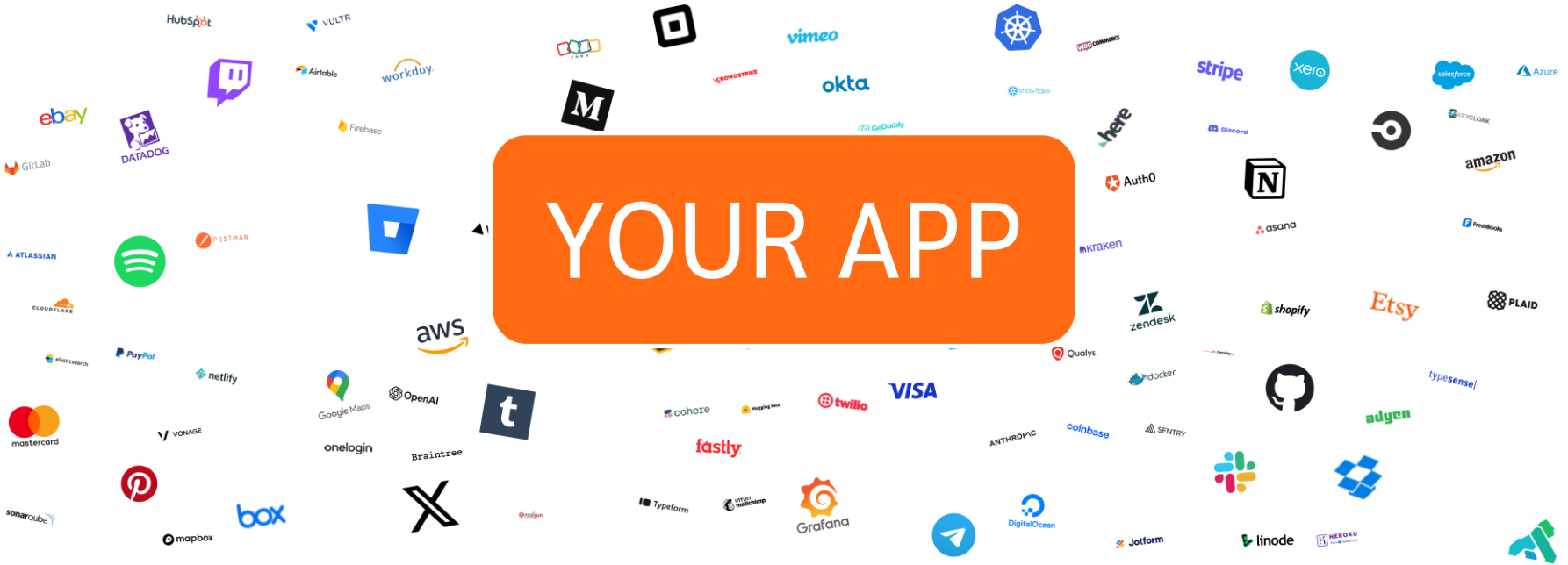
Making Web Services More Accessible

Stark

- **A routing layer** on top of Jarvis
  - Register routes: verb + path → your APL function
- **Parameters arrive pre-parsed**
  - No digging through the URL yourself
- **OpenAPI spec generated** from your routes
  - Served at `/openapi.json`
- **In production soon**
  - DCMS, the video API behind the new dyalog.com



Why should I care?



Why should I care?



Stark Step 1 – Agree the contract

GET `/products/{productId}/price?quantity=20`

PARAMETERS

`productId` path · required
A product id, e.g. NB

`quantity` query · optional
Whole number ≥ 1, default 1

RESPONSES

200	`{ productId, quantity, totalPence }`	
400	`{ message }`	quantity not a whole number ≥ 1
404	`{ message }`	no such product

```
⍝ the data: the catalogue from Exercise 2
Catalogue←(
    (id: 'NB'  ⋄ name: 'Notebook' ⋄ unitPricePence: 400)
    (id: 'PEN' ⋄ name: 'Pen'      ⋄ unitPricePence: 200)
)
```

Design first

Decide what the endpoint promises before writing any code.

Stark Step 2 – Describe the parameters

```
⍝ one namespace per parameter
productId←(name: 'productId' ⋄ in: 'path' ⋄ required: ⊂'true'
    description: 'A product id, e.g. NB'
    schema: (type: 'string'))

quantity←(name: 'quantity' ⋄ in: 'query'
    description: 'How many to price'
    schema: (type: 'integer' ⋄ minimum: 1 ⋄ default: 1))
```

A parameter says where it goes and what type it is

`⊂'true'` is how `⎕JSON` writes true. Path parameters are always required; query parameters are optional unless you say so.

Stark Step 3 – Describe the responses

```
⍝ what comes back, as JSON Schema
Price←(type: 'object' ⋄ properties: (
    productId:  (type: 'string')
    quantity:   (type: 'integer')
    totalPence: (type: 'integer')
))
Error←(type: 'object' ⋄ properties: (message: (type: 'string')))
```

```
⍝ the whole operation
op←(operationId: 'GetPrice'
    summary: 'Price a quantity of one product'
    tags: ('products' ⋄)
    parameters: productId quantity
    responses: (200 Price)(400 Error)(404 Error))
```

operationId names the function in a generated client

A client generated from this spec has `client.products.GetPrice`.

Stark Step 4 – Publish the contract

```
∇ r←GetPrice req      ⍝ not written yet
  req.Fail 501
  r←(message: 'Not implemented yet')
∇

router←Stark.New ⍬
router.Info←(title: 'Catalogue' ⋄ version: '1.0.0')
'/products/{productId}/price' router.Get ('GetPrice' op)
router.Start 8080
```

The spec is live before the code is

`/openapi.json` is already complete. A call returns 501 Not Implemented, but anyone can generate a client now.

Stark Step 5 – Write the function

```
∇ r←GetPrice req;i;n
  i←Catalogue.id⍳⊂req.PathParams.productId
  n←⍎⍎⎕VFI req.QueryParams ⎕VGET ⊂'quantity' '1'
  :If i>≢Catalogue
      req.Fail 404 ⋄ r←(message: 'No such product')
  :ElseIf (n<1)∨n≠⌊n
      req.Fail 400 ⋄ r←(message: 'quantity must be a whole number, 1 or more')
  :Else
      r←(productId: i⊃Catalogue.id ⋄ quantity: n)
      r.totalPence←n×i⊃Catalogue.unitPricePence
  :EndIf
∇
```

Stark doesn't check the contract for you

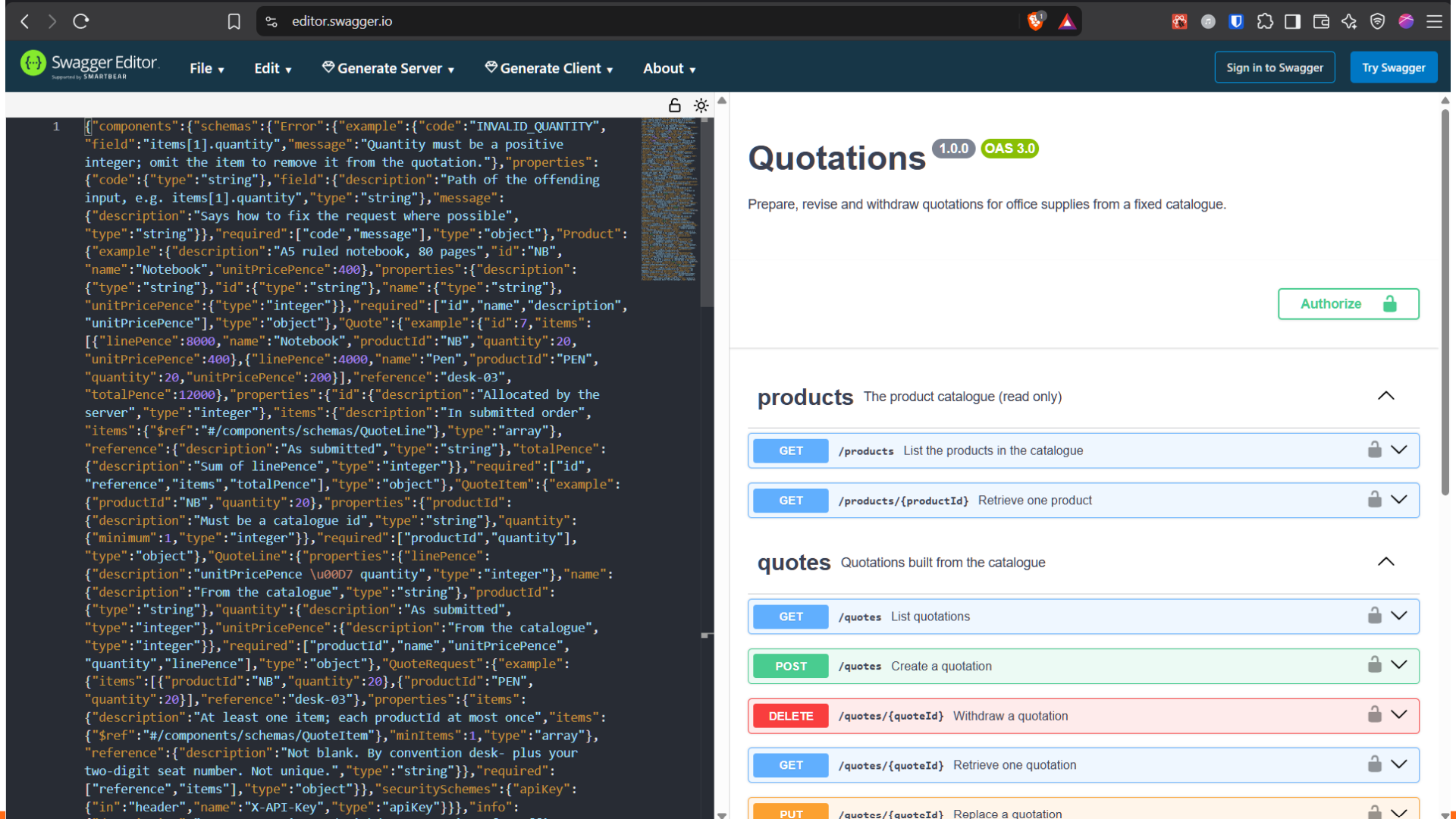
The handler validates `quantity` and picks the status. The server is still running: redefining `GetPrice` is enough.

Stark Step 6 – Check it against the contract

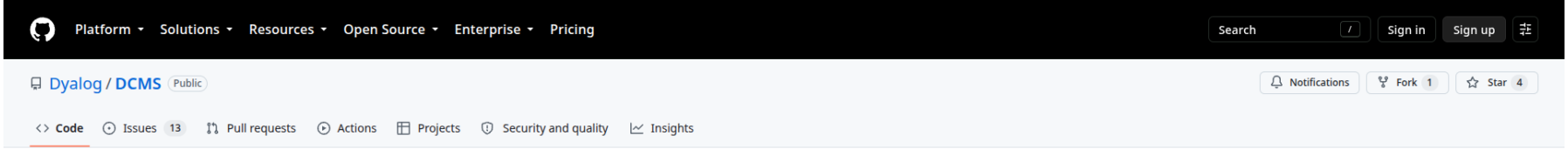
```
$ curl 'localhost:8080/products/NB/price?quantity=20'
{"productId":"NB","quantity":20,"totalPence":8000}          200
$ curl localhost:8080/products/PEN/price
{"productId":"PEN","quantity":1,"totalPence":200}           200
$ curl 'localhost:8080/products/NB/price?quantity=0'
{"message":"quantity must be a whole number, 1 or more"}    400
$ curl localhost:8080/products/XYZ/price
{"message":"No such product"}                               404
```

Every response in the contract, tested

The spec didn't change when the code did: `/openapi.json` is exactly what step 4 published.


editor.swagger.io
Swagger Editor
File
Edit
Generate Server
Generate Client
About
Sign in to Swagger
Try Swagger
{"components":{"schemas":{"Error":{"example":{"code":"INVALID_QUANTITY",
"field":"items[1].quantity","message":"Quantity must be a positive
integer; omit the item to remove it from the quotation."},"properties":
{"code":{"type":"string"},"field":{"description":"Path of the offending
input, e.g. items[1].quantity","type":"string"},"message":
{"description":"Says how to fix the request where possible",
"type":"string"}},"required":["code","message"],"type":"object"},"Product":
{"example":{"description":"A5 ruled notebook, 80 pages","id":"NB",
"name":"Notebook","unitPricePence":400},"properties":{"description":
{"type":"string"},"id":{"type":"string"},"name":{"type":"string"},
"unitPricePence":{"type":"integer"}},"required":["id","name","description",
"unitPricePence"],"type":"object"},"Quote":{"example":{"id":7,"items":
[{"linePence":8000,"name":"Notebook","productId":"NB","quantity":20,
"unitPricePence":400},{"linePence":4000,"name":"Pen","productId":"PEN",
"quantity":20,"unitPricePence":200}],"reference":"desk-03",
"totalPence":12000},"properties":{"id":{"description":"Allocated by the
server","type":"integer"},"items":{"description":"In submitted order",
"items":{"$ref":"#/components/schemas/QuoteLine"},"type":"array"},
"reference":{"description":"As submitted","type":"string"},"totalPence":
{"description":"Sum of linePence","type":"integer"}},"required":["id",
"reference","items","totalPence"],"type":"object"},"QuoteItem":{"example":
{"productId":"NB","quantity":20},"properties":{"productId":
{"description":"Must be a catalogue id","type":"string"},"quantity":
{"minimum":1,"type":"integer"}},"required":["productId","quantity"],
"type":"object"},"QuoteLine":{"properties":{"linePence":
{"description":"unitPricePence \u00D7 quantity","type":"integer"},"name":
{"description":"From the catalogue","type":"string"},"productId":
{"type":"string"},"quantity":{"description":"As submitted",
"type":"integer"},"unitPricePence":{"description":"From the catalogue",
"type":"integer"}},"required":["productId","name","unitPricePence",
"quantity","linePence"],"type":"object"},"QuoteRequest":{"example":
{"items":[{"productId":"NB","quantity":20},{"productId":"PEN",
"quantity":20}],"reference":"desk-03"},"properties":{"items":
{"description":"At least one item; each productId at most once","items":
{"$ref":"#/components/schemas/QuoteItem"},"minItems":1,"type":"array"},
"reference":{"description":"Not blank. By convention desk- plus your
two-digit seat number. Not unique.","type":"string"}},"required":
["reference","items"],"type":"object"}},"securitySchemes":{"apiKey":
{"in":"header","name":"X-API-Key","type":"apiKey"}}},"info":
Quotations
1.0.0
OAS 3.0
Prepare, revise and withdraw quotations for office supplies from a fixed catalogue.
Authorize
products
The product catalogue (read only)
GET /products List the products in the catalogue
GET /products/{productId} Retrieve one product
quotes
Quotations built from the catalogue
GET /quotes List quotations
POST /quotes Create a quotation
DELETE /quotes/{quoteId} Withdraw a quotation
GET /quotes/{quoteId} Retrieve one quotation
PUT /quotes/{quoteId} Replace a quotation

For Example, DCMS



Making Web Services More Accessible

```
Public-facing API documentation: Swagger UI at / driven by /openapi.json.
'/'                          router.Get('WEB.Index'  (WEB.Index'⍠SPEC'))
'/swagger/swagger-ui.css'    router.Get('WEB.Static' (WEB.Static'CSS⍠SPEC'))
'/swagger/theme.css'         router.Get('WEB.Static' (WEB.Static'THEME⍠SPEC'))
'/swagger/swagger-ui-bundle.js' router.Get('WEB.Static' (WEB.Static'JS⍠SPEC'))

'/version'                          router.Get('Version'                   (Version'⍠SPEC'))
'/videos/{youtube_id}/recommendations' router.Get('QUERY.VIDEOS.Recommendations'(QUERY.VIDEOS.Recommendations'⍠SPEC'))
'/videos/{youtube_id}'              router.Get('QUERY.VIDEOS.Single'       (QUERY.VIDEOS.Single'⍠SPEC'))
'/videos'                           router.Get('QUERY.VIDEOS.Query'        (QUERY.VIDEOS.Query'⍠SPEC'))
'/presenters'                       router.Get('QUERY.Presenters'          (QUERY.Presenters'⍠SPEC'))
'/events'                           router.Get('QUERY.Events'              (QUERY.Events'⍠SPEC'))

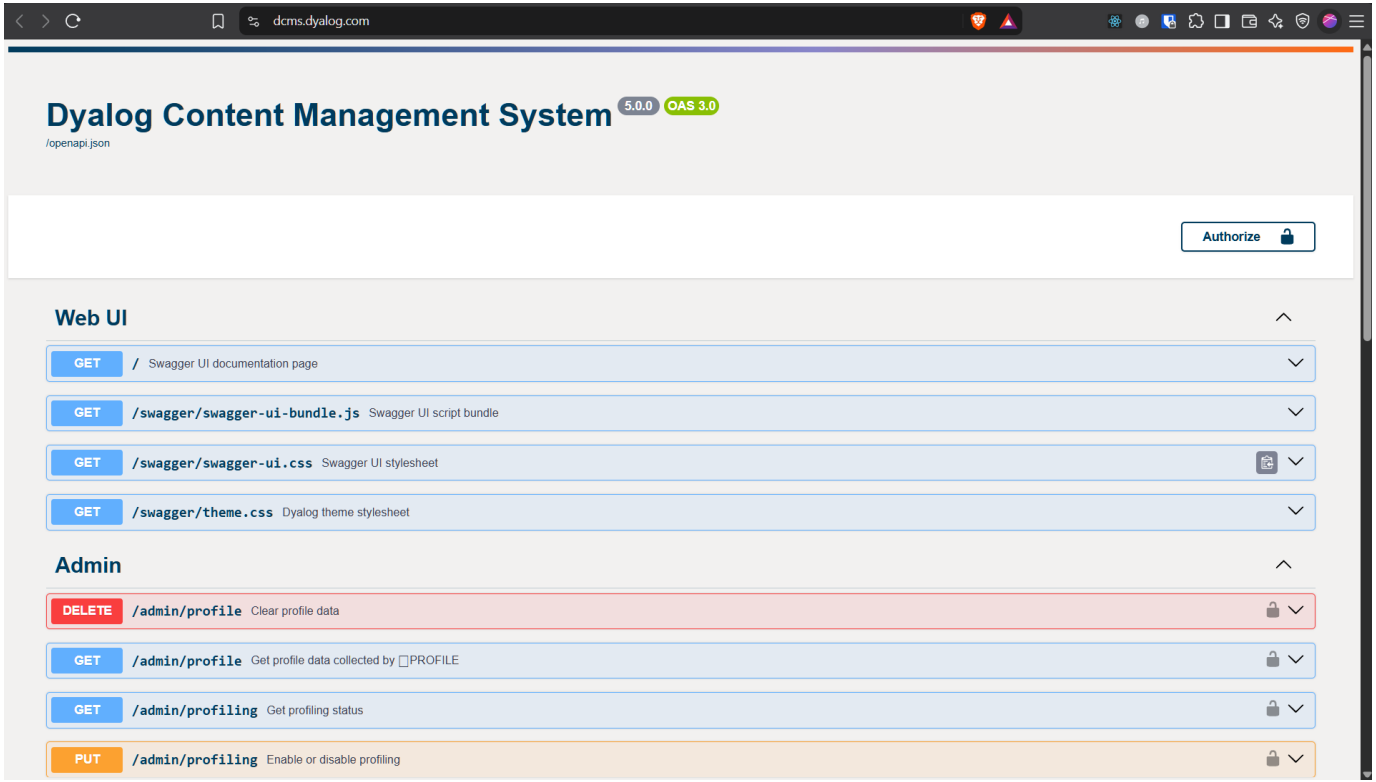
'/schema/{table}'    router.Get   ('CRUD.Schema' (CRUD.Schema'⍠SPEC'))
'/crud'              router.Get   ('CRUD.Tables' (CRUD.Tables'⍠SPEC'))
'/crud/{table}'      router.Get   ('CRUD.Read'   (CRUD.Read'⍠SPEC'))
'/crud/{table}'      router.Post  ('CRUD.Create' (CRUD.Create'⍠SPEC'))
'/crud/{table}/{id}' router.Get   ('CRUD.Read'   (CRUD.Read'ID⍠SPEC'))
'/crud/{table}/{id}' router.Patch ('CRUD.Update' (CRUD.Update'⍠SPEC'))
'/crud/{table}/{id}' router.Delete('CRUD.Delete' (CRUD.Delete'⍠SPEC'))

'/admin/refresh'        router.Post('ADMIN.Refresh'       (ADMIN.Refresh'⍠SPEC'))
'/admin/refresh/status' router.Get ('ADMIN.RefreshStatus' (ADMIN.RefreshStatus'⍠SPEC'))
'/admin/wp-push'        router.Post('ADMIN.WPPush'        (ADMIN.WPPush'⍠SPEC'))

'/admin/profiling'   router.Get   ('PROFILING.Status' (PROFILING.Status'⍠SPEC'))
'/admin/profiling'   router.Put   ('PROFILING.Toggle' (PROFILING.Toggle'⍠SPEC'))
'/admin/profile'     router.Get   ('PROFILING.Data'   (PROFILING.Data'⍠SPEC'))
'/admin/profile'     router.Delete('PROFILING.Clear'  (PROFILING.Clear'⍠SPEC'))
PROFILING.status←(on: 0 ⋄ since: ⊂'null' ⋄ last_cleared: ⊂'null')
```

```
r←Read req;GBL;data;fields;id;idx;parameters;pg;ppg;responses;schemas;table
⍝ Handles GET /crud/{table}; lists paginated rows or fetches one by id

:If 2=⎕NC'req'
    (schemas responses parameters)←##.SPEC.(schemas responses parameters)
:EndIf
:If req≡'⎕SPEC'
    r←(
        tags:,⊂'Table CRUD' ⋄ summary:'List rows from table' ⋄ operationId:'listRows'
        parameters:(parameters.Page ⋄ parameters.PerPage)
        responses:(
            Δ200:(
                description:'Paginated list of table rows.'
                content:(Δapplication Δ47 Δjson:(schema:(allOf:(
                                schemas.pagination_envelope
                                (
                                    type:'object'
                                    properties:(data:(type:'array' ⋄ items:schemas.crud_item))
                                )
                            ))))
            )
            Δ401:responses.unauthorized
            Δ404:responses.not_found
        )
    )
    :Return
```


dcms.dyalog.com
Dyalog Content Management System
5.0.0
OAS 3.0
/openapi.json
Authorize
Web UI
GET / Swagger UI documentation page
GET /swagger/swagger-ui-bundle.js Swagger UI script bundle
GET /swagger/swagger-ui.css Swagger UI stylesheet
GET /swagger/theme.css Dyalog theme stylesheet
Admin
DELETE /admin/profile Clear profile data
GET /admin/profile Get profile data collected by ⎕PROFILE
GET /admin/profiling Get profiling status
PUT /admin/profiling Enable or disable profiling

```
{
  "components": {
    "securitySchemes": {
      "ApiKeyAuth": {
        "in": "header",
        "name": "X-API-Key",
        "type": "apiKey"
      }
    }
  },
  "info": {
    "title": "Dyalog Content Management System",
    "version": "5.0.0"
  },
  "openapi": "3.0.3",
  "paths": {
    "/": {
      "get": {
        "operationId": "getDocs",
        "responses": {
          "200": {
            "description": "HTML page hosting the Swagger UI"
          }
        },
        "security": [],
        "summary": "Swagger UI documentation page",
        "tags": [
          "Web UI"
        ]
      }
    },
    "/admin/profile": {
      "delete": {
        "operationId": "clearProfile",
        "responses": {
          "204": {
            "description": "Profile data cleared. No response body."
          },
          "401": {
            "content": {
              "application/json": {
                "schema": {
```

Conclusion

- Web Services Overview
- Consuming
  - Dyalog OpenAPI Client Generator
- Serving
  - Stark