

Postman hebt ab – und nun?

Alternative API-Clients im Überblick



> whoami



Andreas Siegel

Entwickler | Architekt | Allesforscher



andreas.siegel@pentacor.de

linktr.ee/andreassiegel



PENTACOR
ARCHITECTING
YOUR BUSINESS



www.pentacor.de



+49 (0)1590 1395830



info@pentacor.de



Schönherrstr. 8
09113 Chemnitz
Germany

UNSERE
THEMEN



IT-Architektur



Cloud-Native Software



APIs & Datenprodukte

Anwendungsfälle für einen API-Client



API-EXPLORATION



SUPPORT &
TROUBLESHOOTING



MAINTENANCE

Postman im Überblick

A screenshot of the Postman API client interface. The main window shows a GET request configuration for the endpoint `https://api.notion.com/v1/databases/id`. The request is configured with a query parameter `id` with the value `{{DATABASE_ID}}`. The response is shown in the 'Body' tab, displaying a JSON object for a Notion database object. The response status is 200 OK, with a response time of 1033ms and a size of 120KB. The interface includes a sidebar with a collection tree, a top navigation bar, and a right-hand panel with documentation and authorization details.

Home Workspaces Reports Explore Search Postman

Notion's Public Workspace New Import Overview GET Retrieve a database X + ... Notion API

Collection + ...

Environment

APIs

Mock Servers

Monitors

Flows

History

Notion API

Users

Databases

GET Retrieve a database

200 Success- Retri...

POST Query a database

POST Create a database

PATCH Update a database

Pages

Users

Blocks

Comments

Notion API / Databases / Retrieve a database Save

GET https://api.notion.com/v1/databases/id Send

Params Auth Headers(10) Body Pre-req. Tests Settings Cookies

Query params

KEY	VALUE	DESCRIPTION	Bulk edit
Key	Value	Description	

Path variables

KEY	VALUE	DESCRIPTION	Bulk edit
id	{{DATABASE_ID}}	Required. Enter database id.	

Body 200 OK 1033ms 120KB Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "Publisher": {
3     "id": "%3E%24Pb",
4     "name": "Publisher",
5     "type": "select",
6     "select": {
7       "options": [
8         {
9           "id": "c5ee409a-f307-4176-99ee-6e424fa89afa",
10          "name": "NYT",
11          "color": "default"
12        }
13      ]
14     }
15   }
16 }
```

Documentation

https://api.notion.com/v1/databases/id

Retrieves a database object using the ID specified in the request path.

Authorization Bearer Token

This request is using an authorization helper from collection Notion API

Request Header

Notion-Version 22-02-22

Path Variables

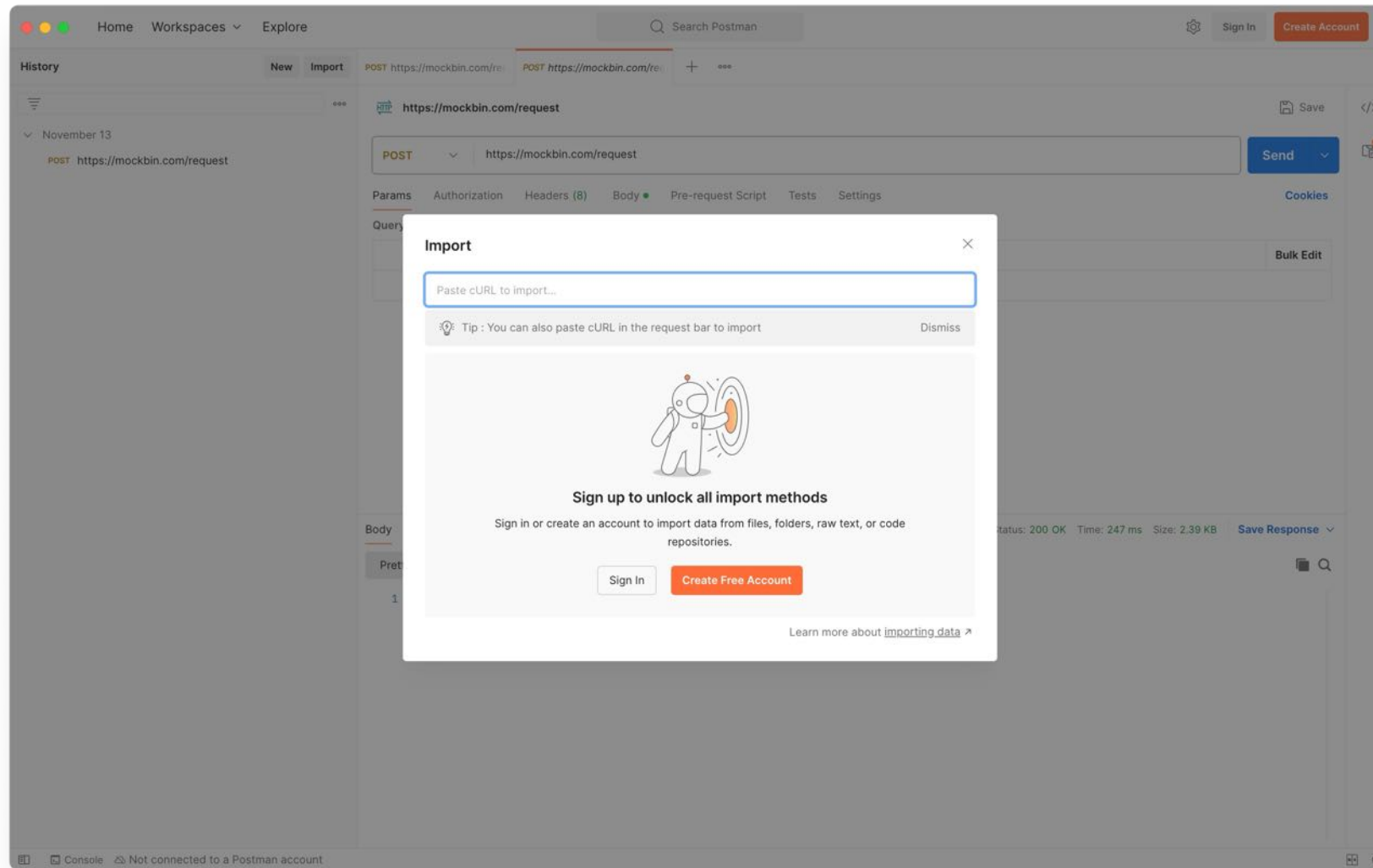
id {{DATABASE_ID}}

Required. Enter database id.

View complete collection documentation ->

Find and Replace Console Cookies Capture Requests Bootcamp Runner Trash

Postman Lightweight API Client



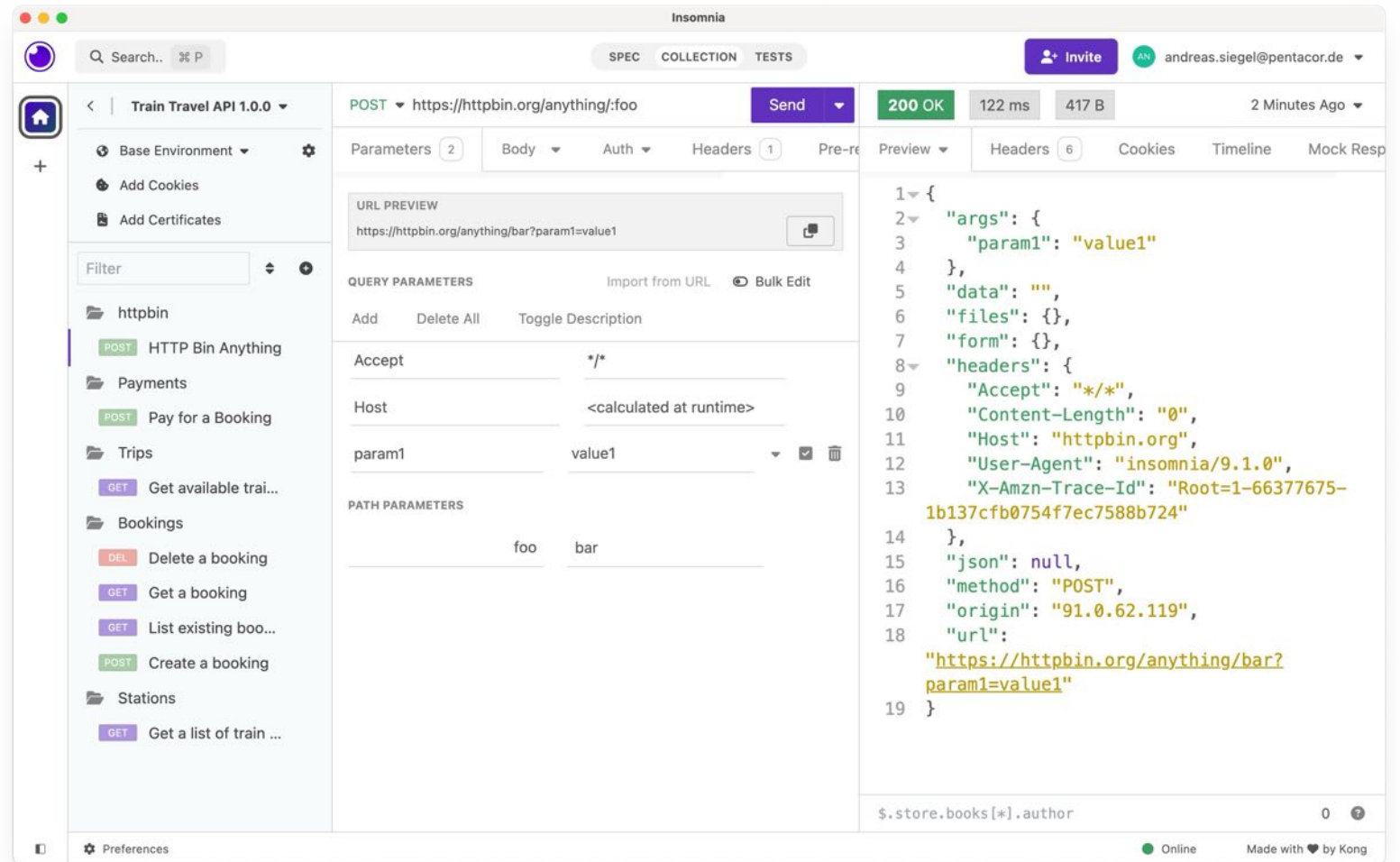
Ausgewählte Alternative API-Clients

Welche anderen Tools gibt es für die Arbeit mit APIs?

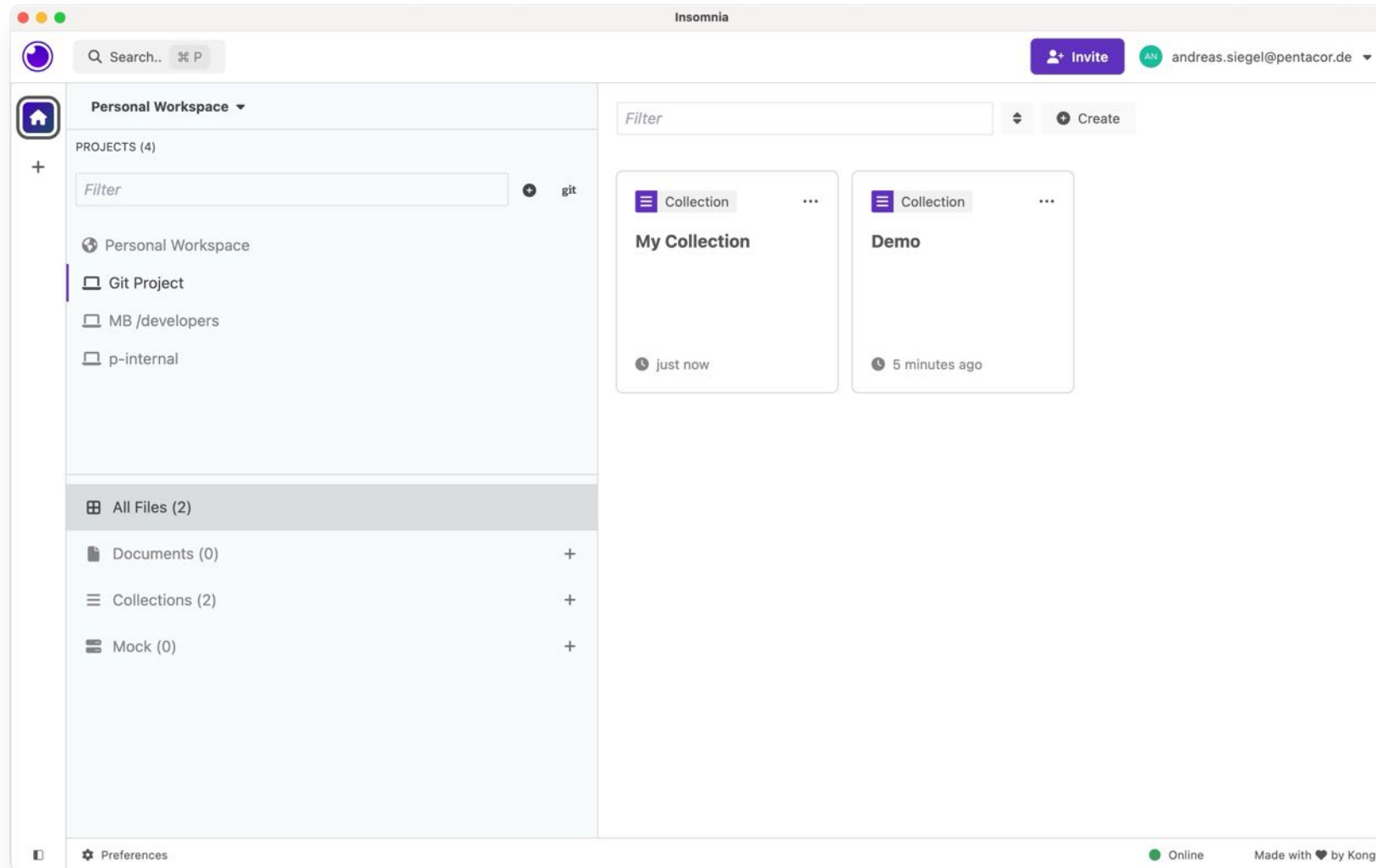


Insomnia

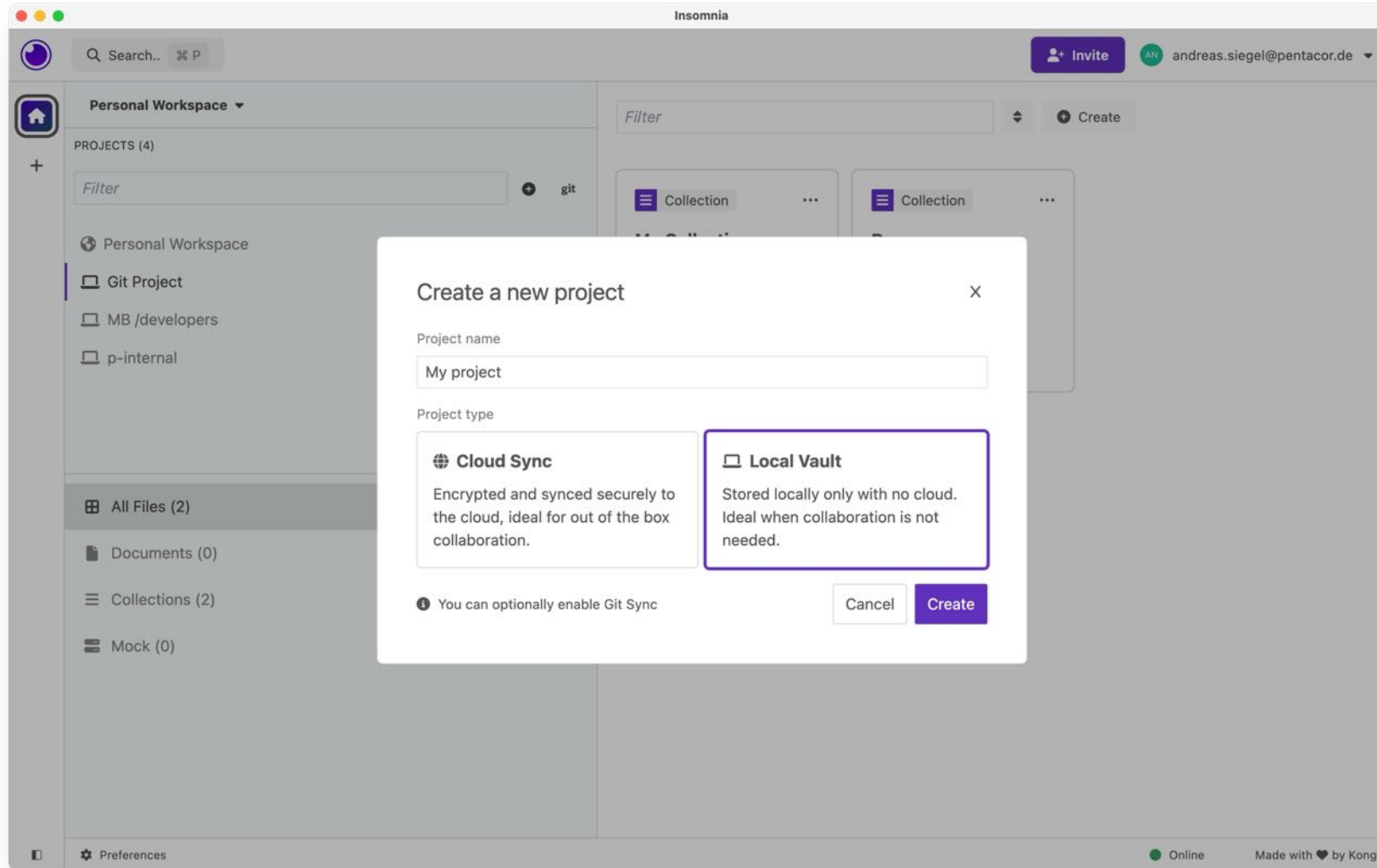
insomnia.rest



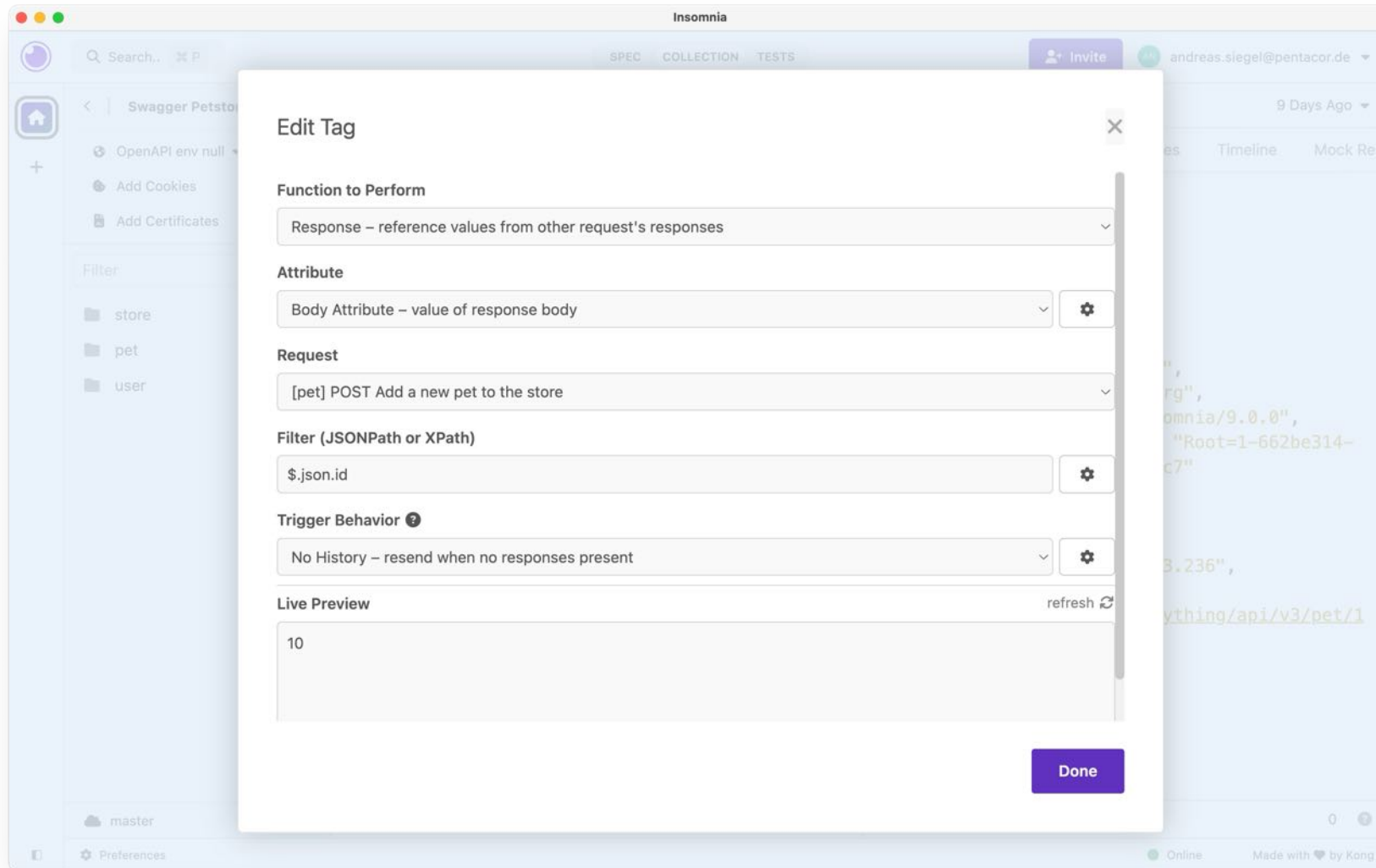
Insomnia: Workspace(s), Projekte und Collections



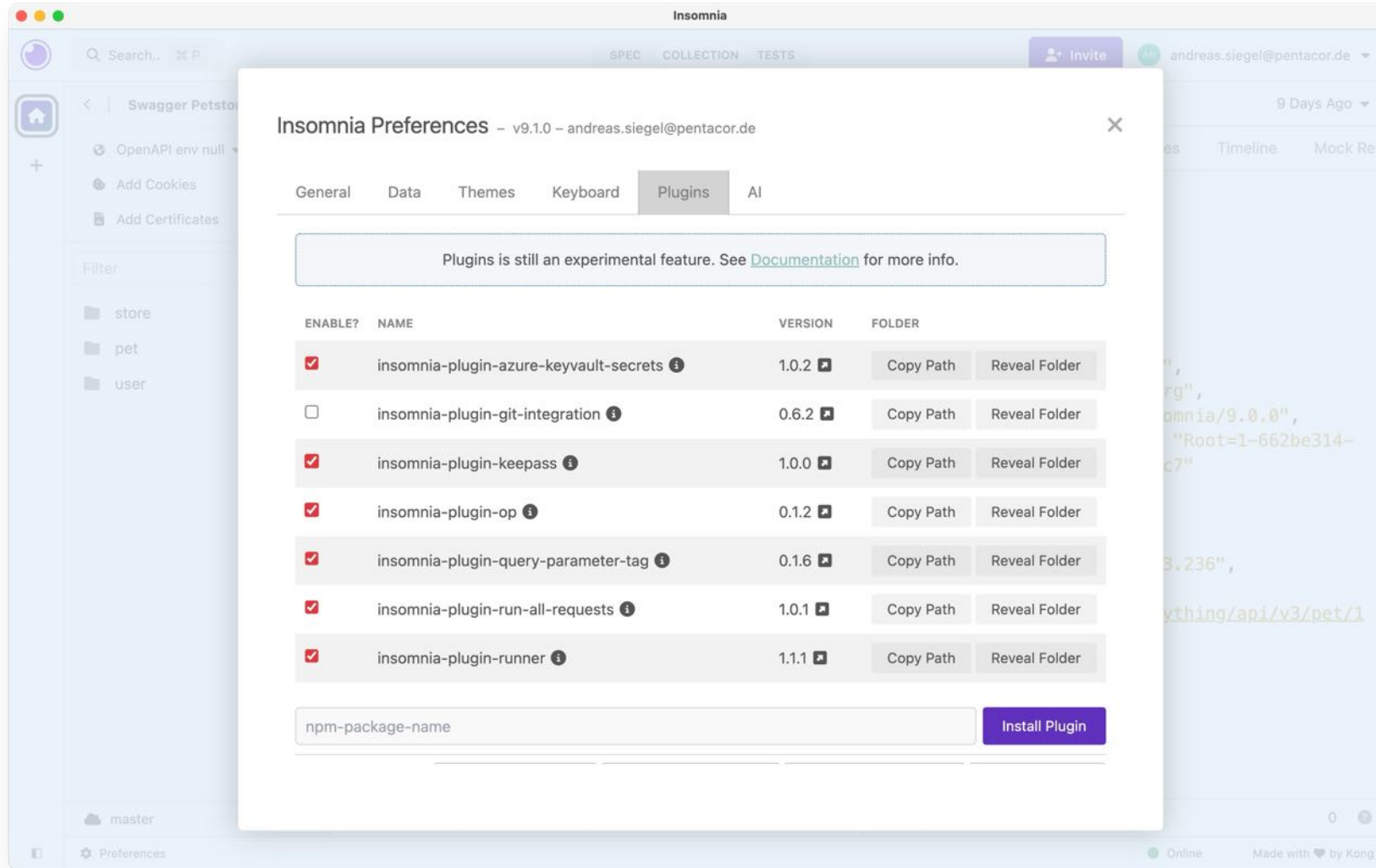
Insomnia: Cloud Sync & Lokale Datenspeicherung



Insomnia: Functions & Request Chaining

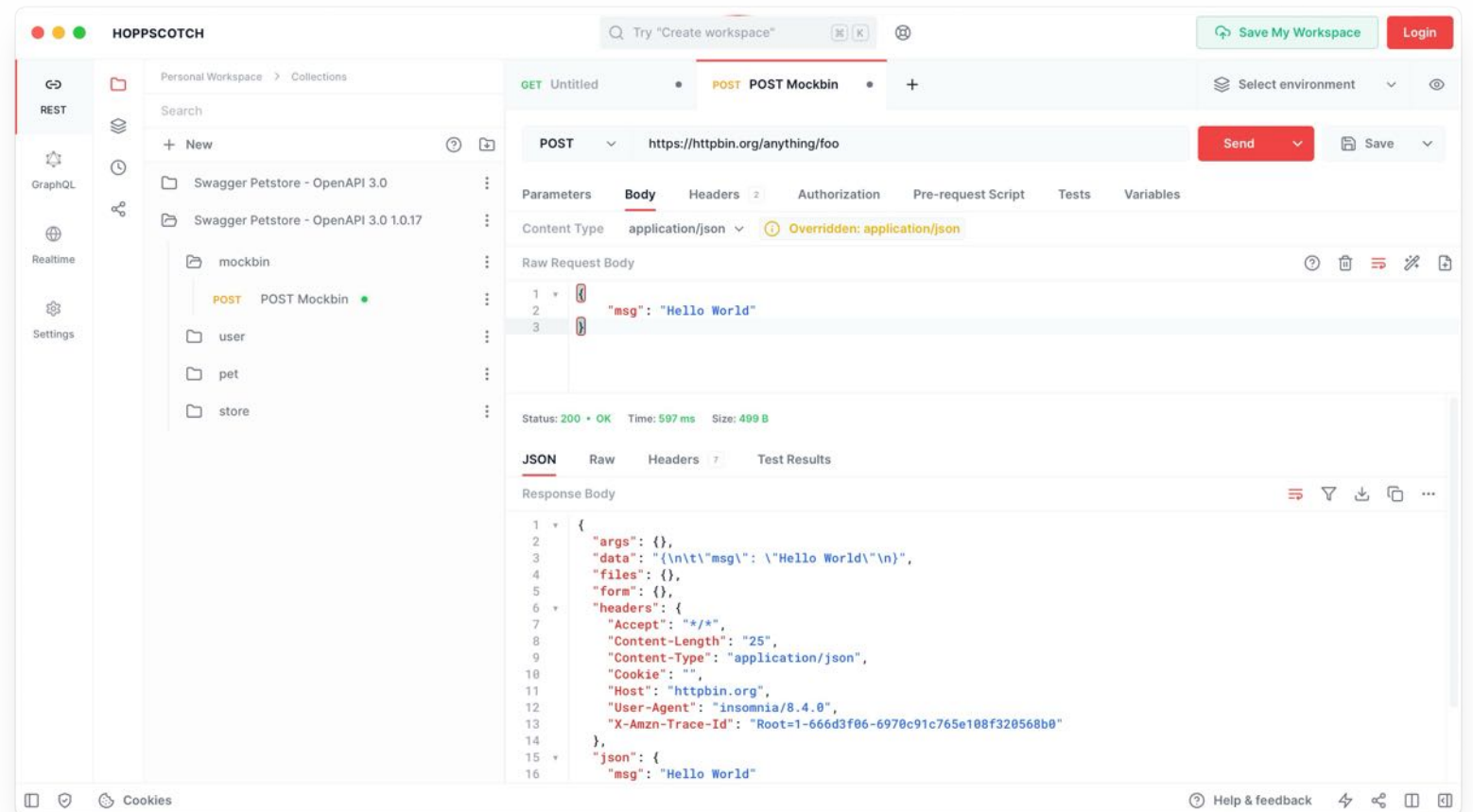


Insomnia: Plugins

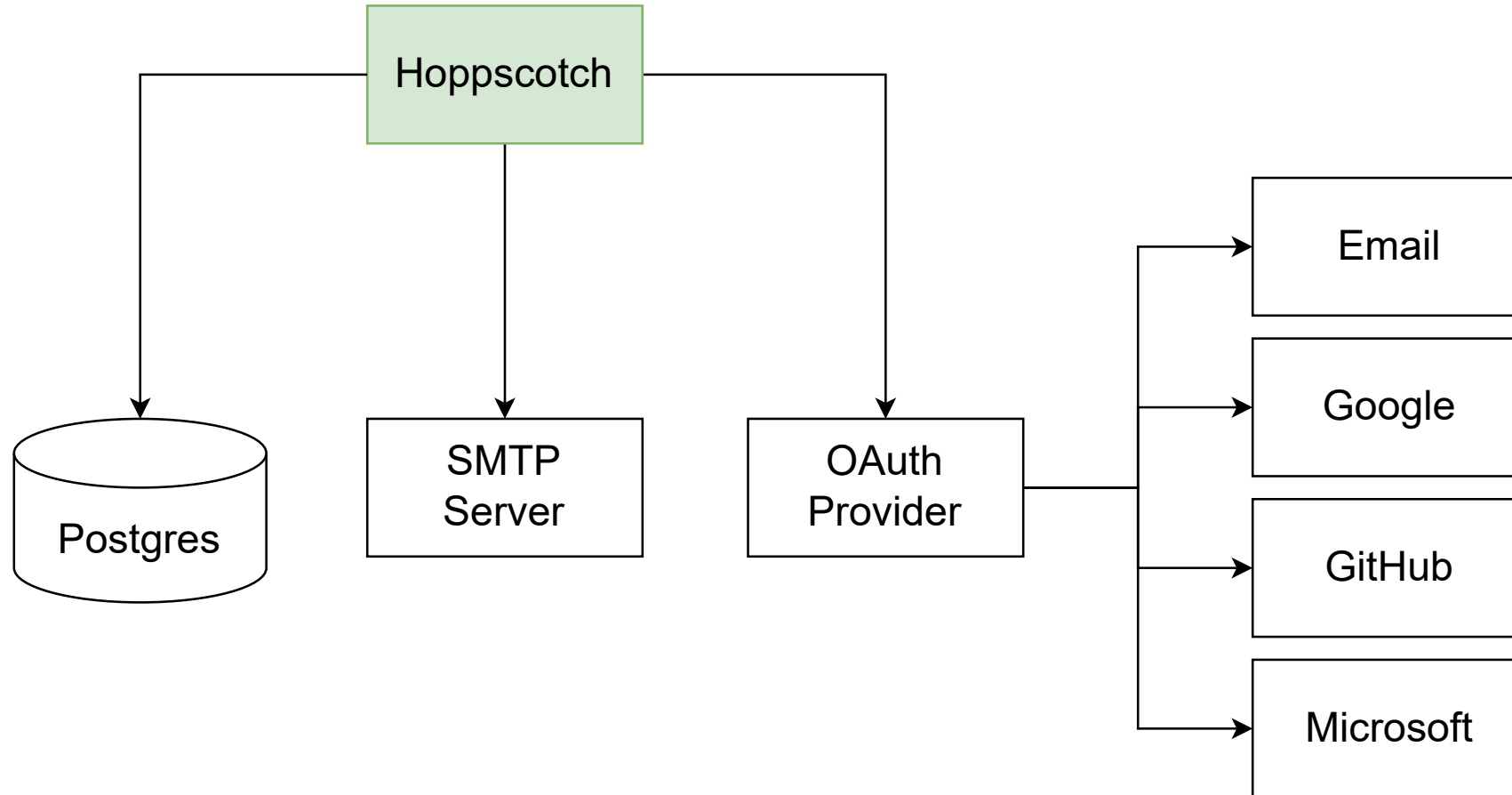


Hoppscotch

hoppscotch.com

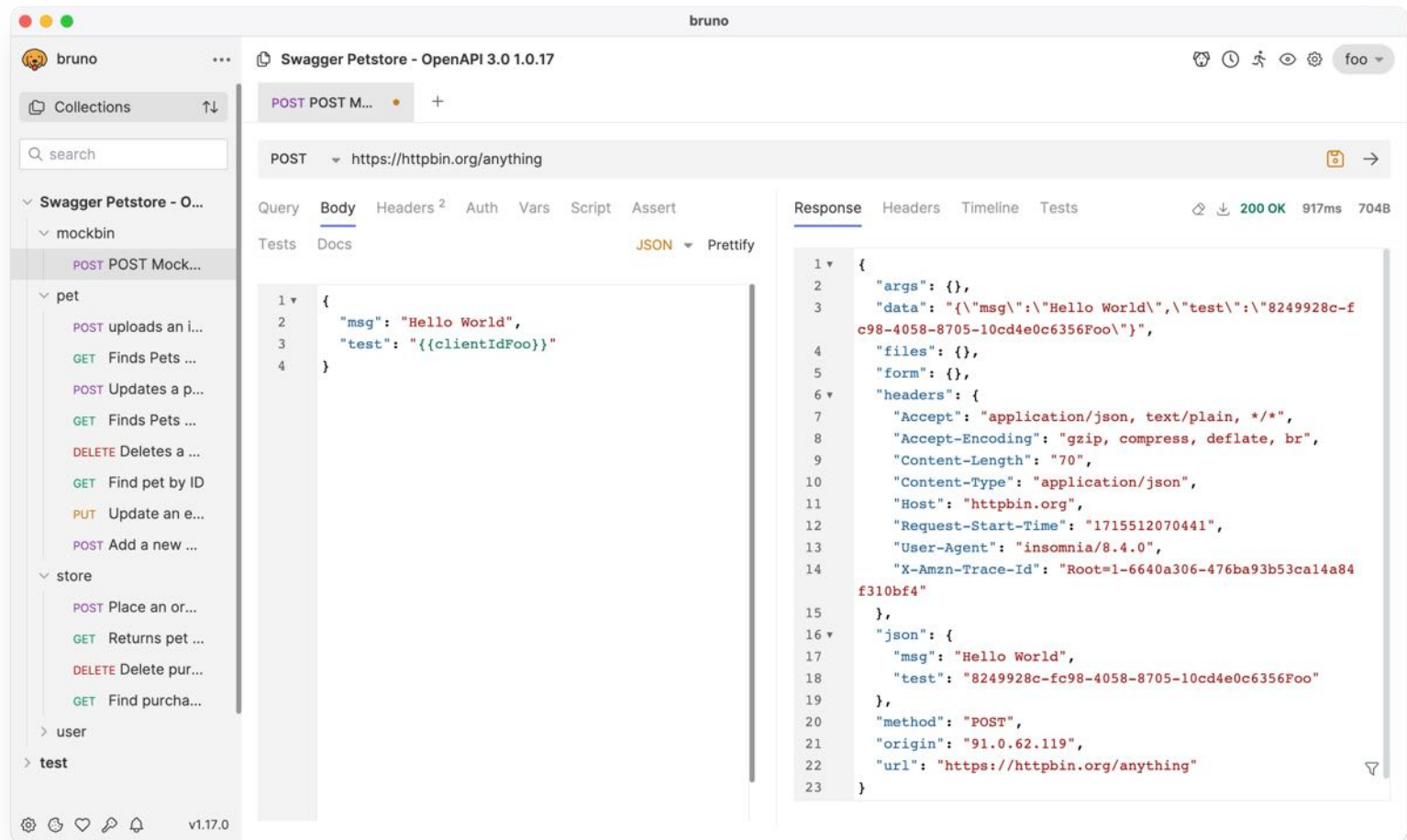


Hoppscotch: Self-Hosting

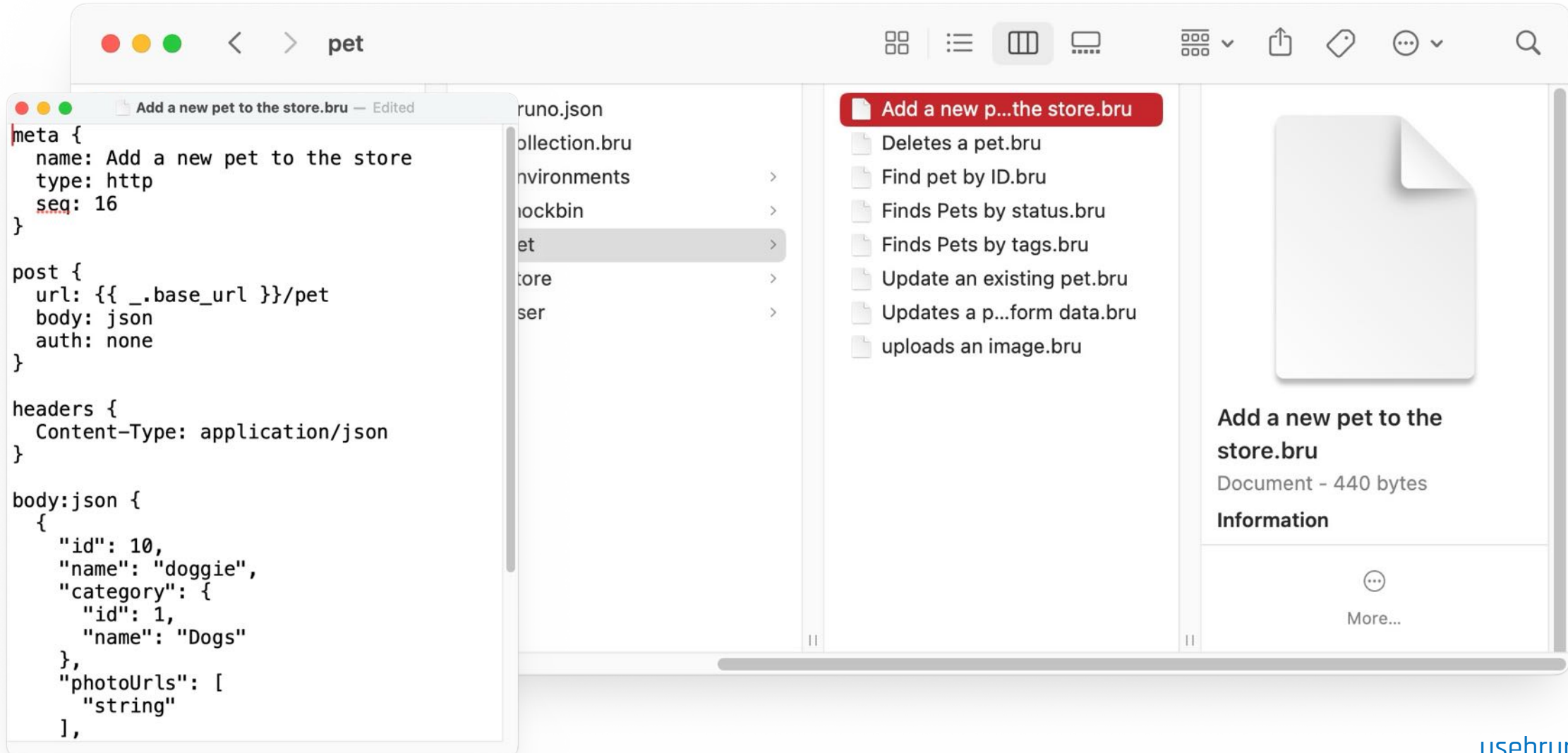


Bruno

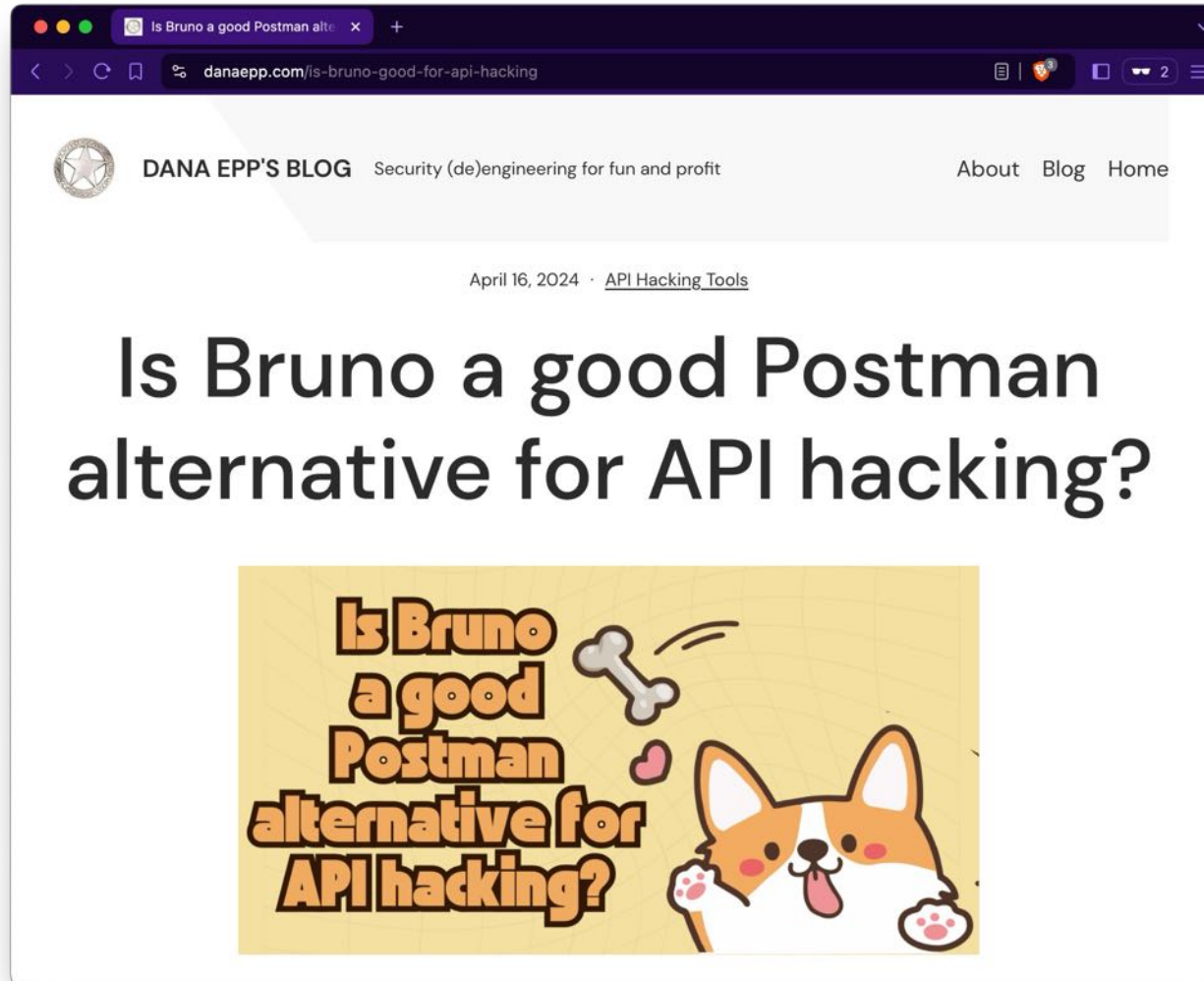
usebruno.com



Bruno: Lokale Datenspeicherung



Bruno: Ready for Production?

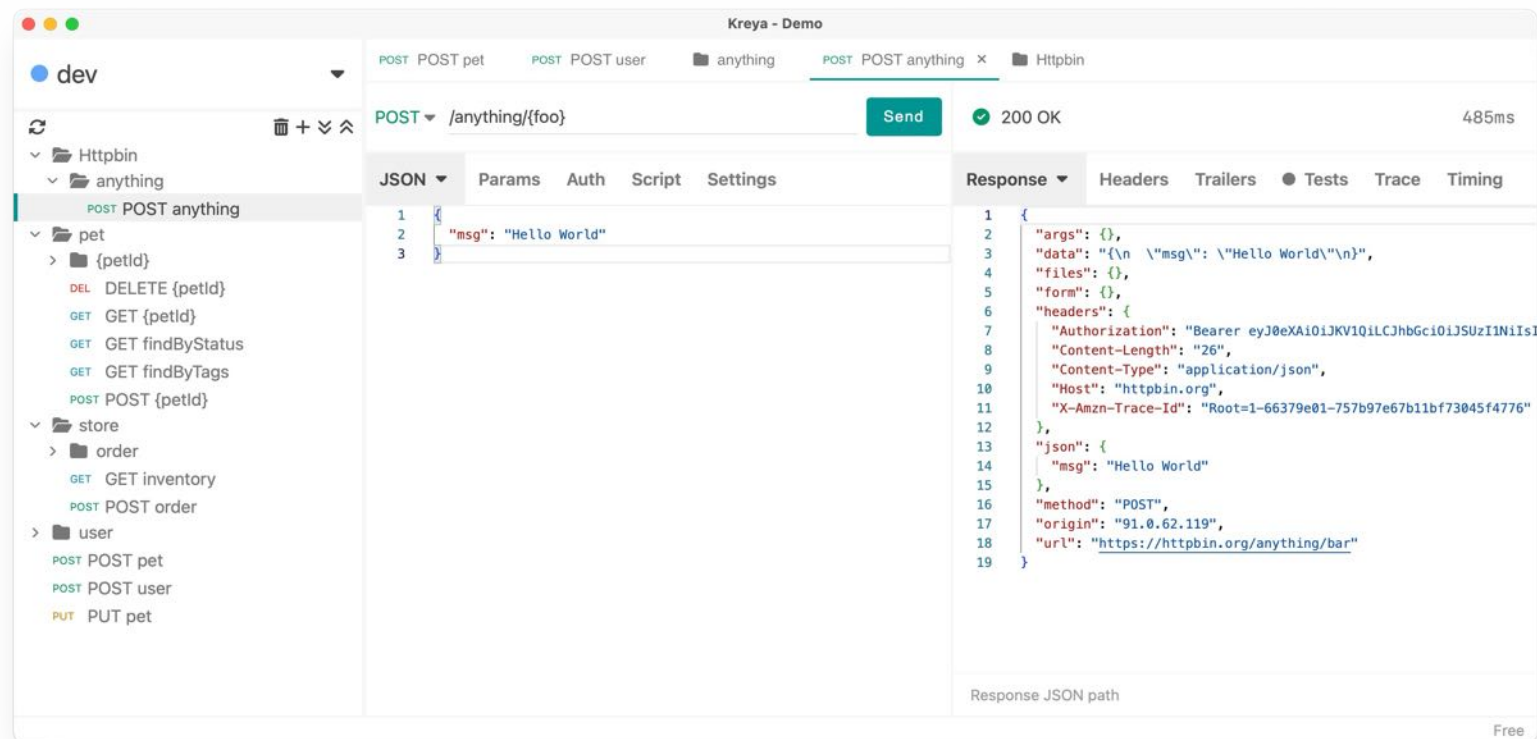


[https://danaepp.com/
is-bruno-good-for-api-hacking](https://danaepp.com/is-bruno-good-for-api-hacking)



Kreya

kreya.app



Kreya: Importer



Kreya - Demo

Project settings

Back

Environments Authentications **Importers** Certificates Default settings

New importer

open-api-file-importer

Name* open-api-file-importer

Type* REST OpenAPI file

Create an operation for each API method

Project subdirectory to create missing operations Pet Store

Prefix directory to trim when creating missing operations

Path to OpenAPI document* ../../openapi.json

Delete Duplicate Save

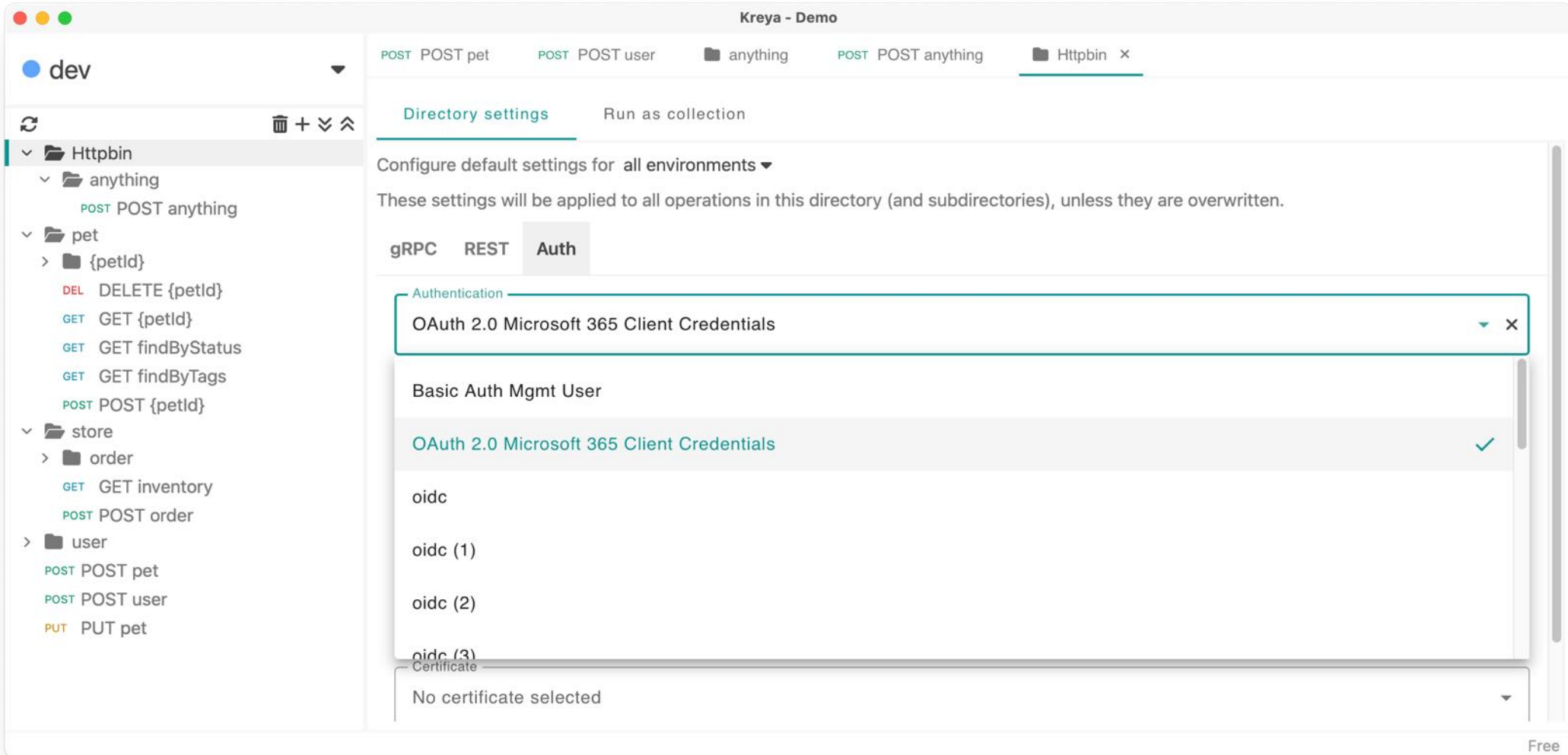
Kreya: Konfiguration

The screenshot displays the Kreya application interface. On the left, a sidebar shows a directory tree under 'dev' with folders like 'Httpbin', 'anything', 'pet', 'store', and 'user'. The main panel is titled 'Kreya - Demo' and shows 'Directory settings' for the 'Httpbin' directory. It includes tabs for 'gRPC', 'REST', and 'Auth', with 'REST' selected. The settings are for 'all environments' and include:

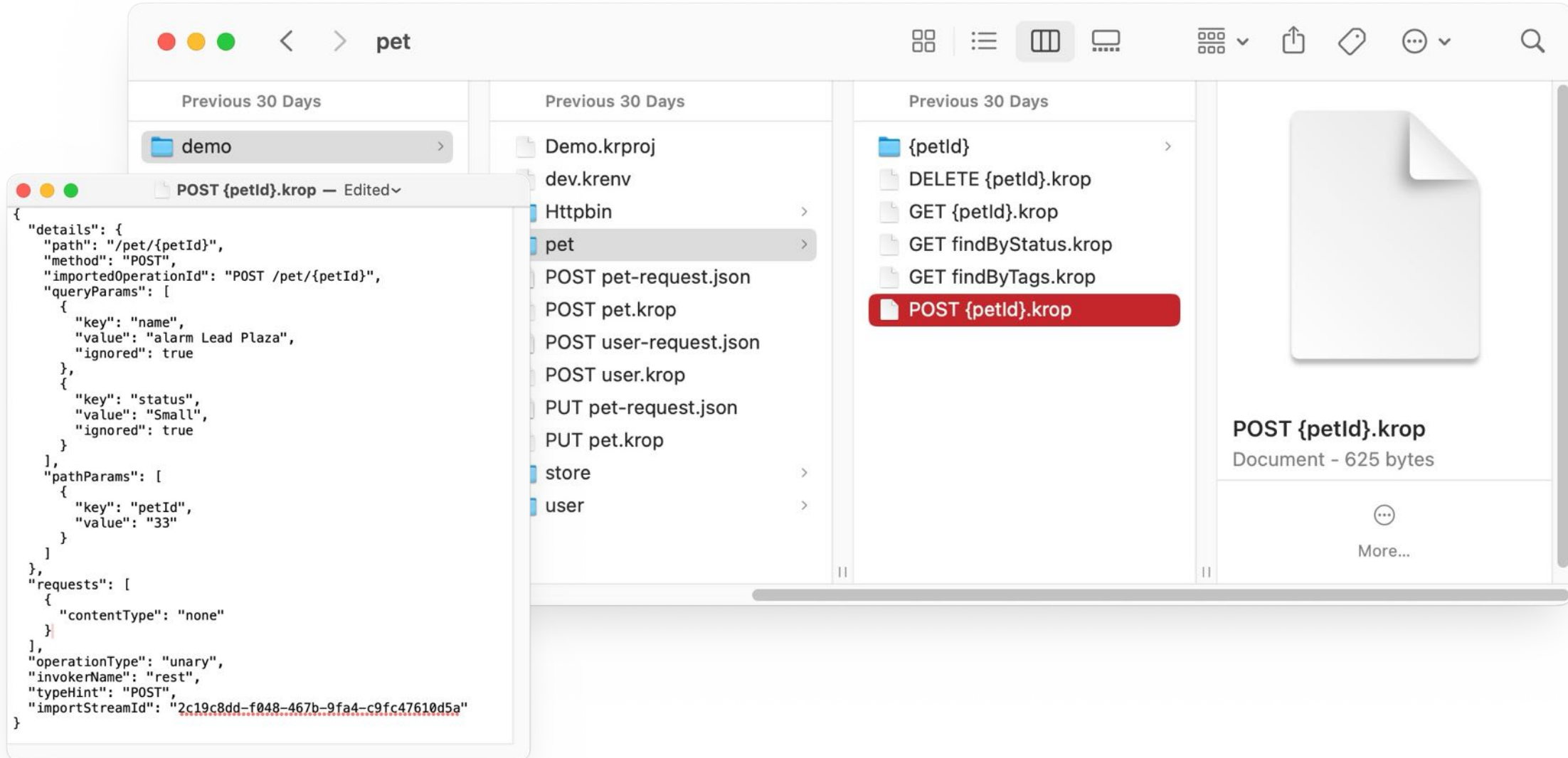
- Endpoint: `https://httpbin.org`
- Server certificate (TLS/SSL) validation: Enabled
- Automatically follow redirects: Enabled
- HTTP version: 1.1
- HTTP version policy: Version or higher

At the bottom, there is a 'Query params' section and a 'Free' label in the bottom right corner.

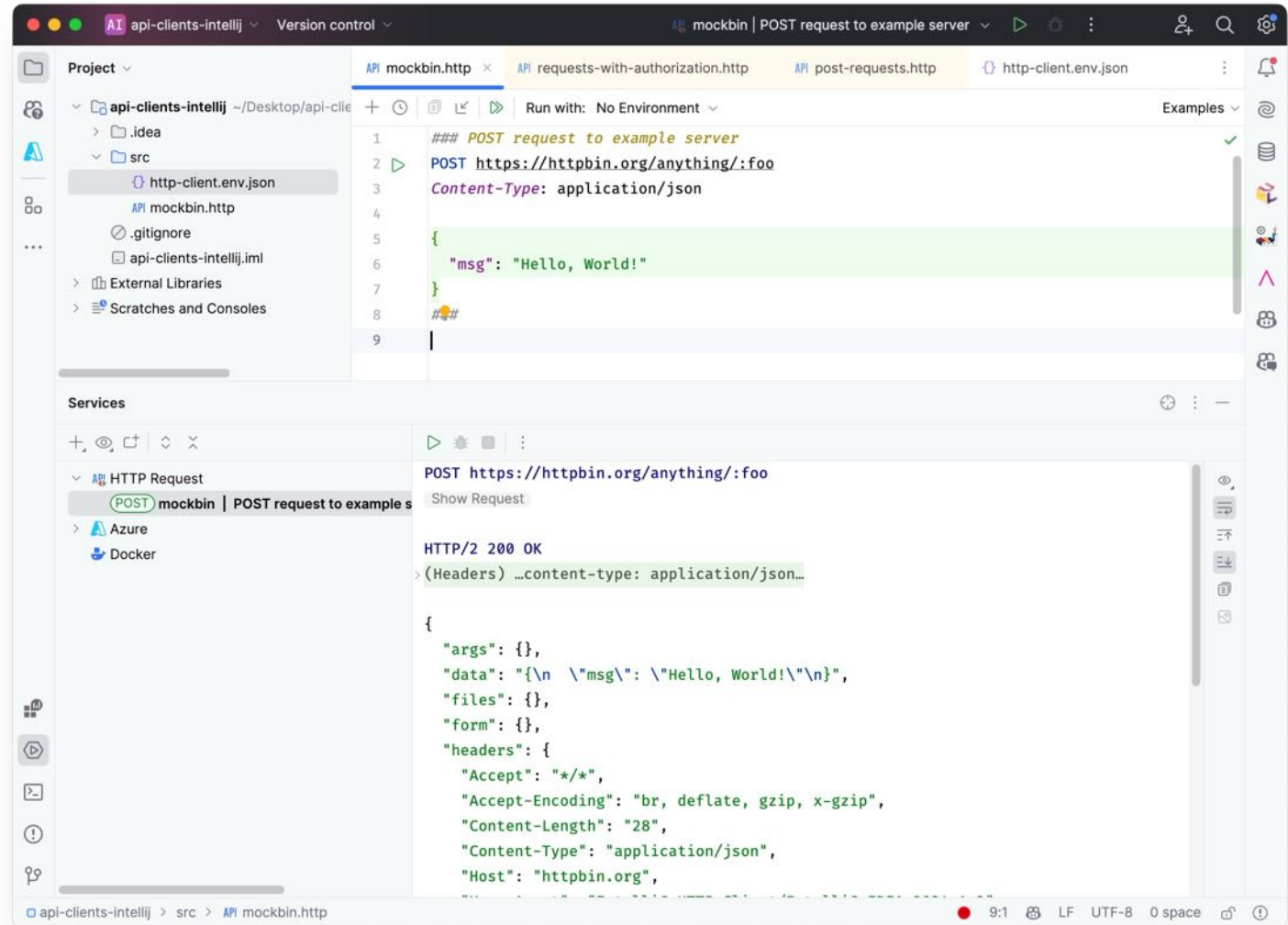
Kreya: Authentifizierung



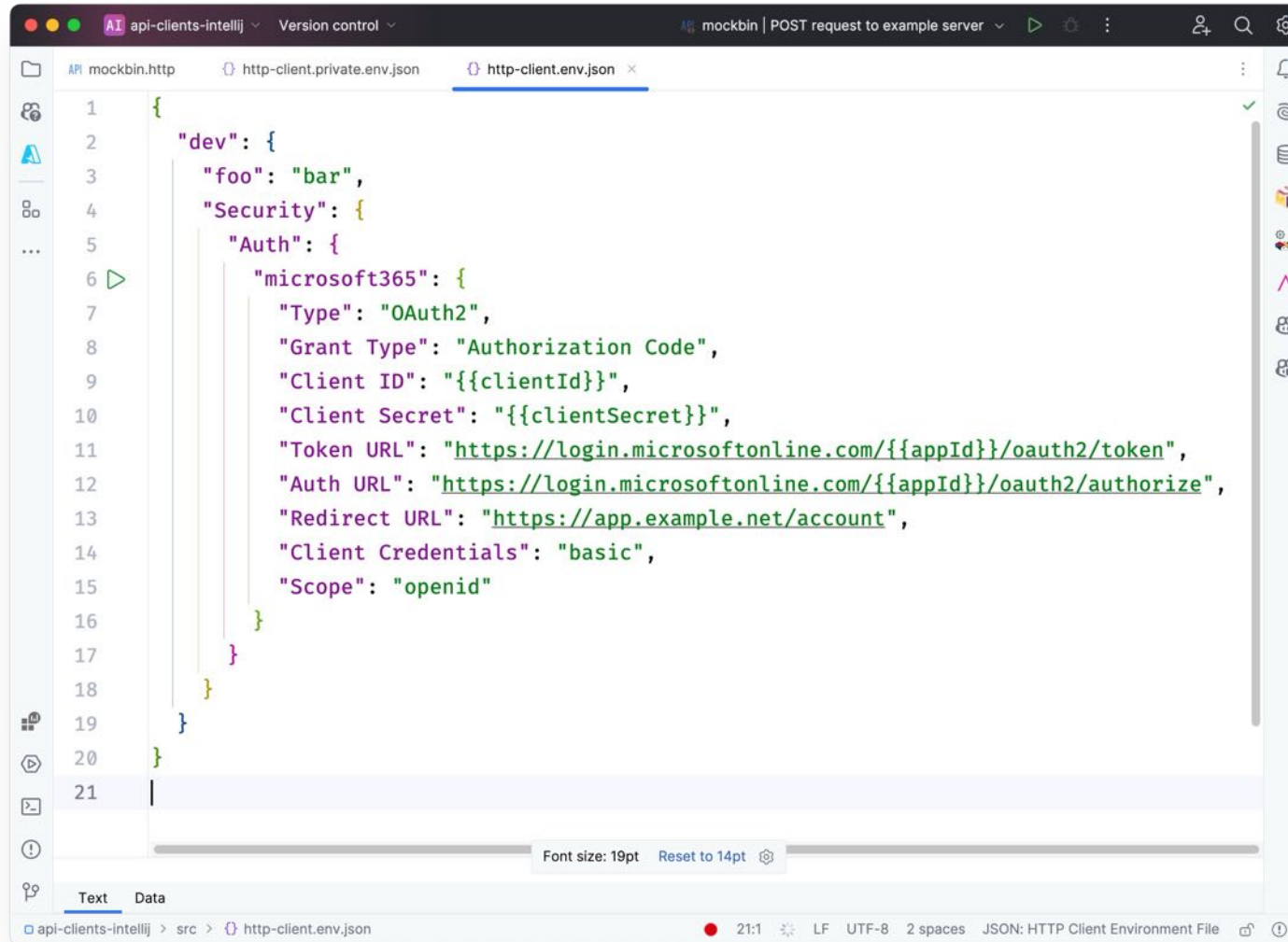
Kreya: Lokale Datenspeicherung



IntelliJ IDEA HTTP Client

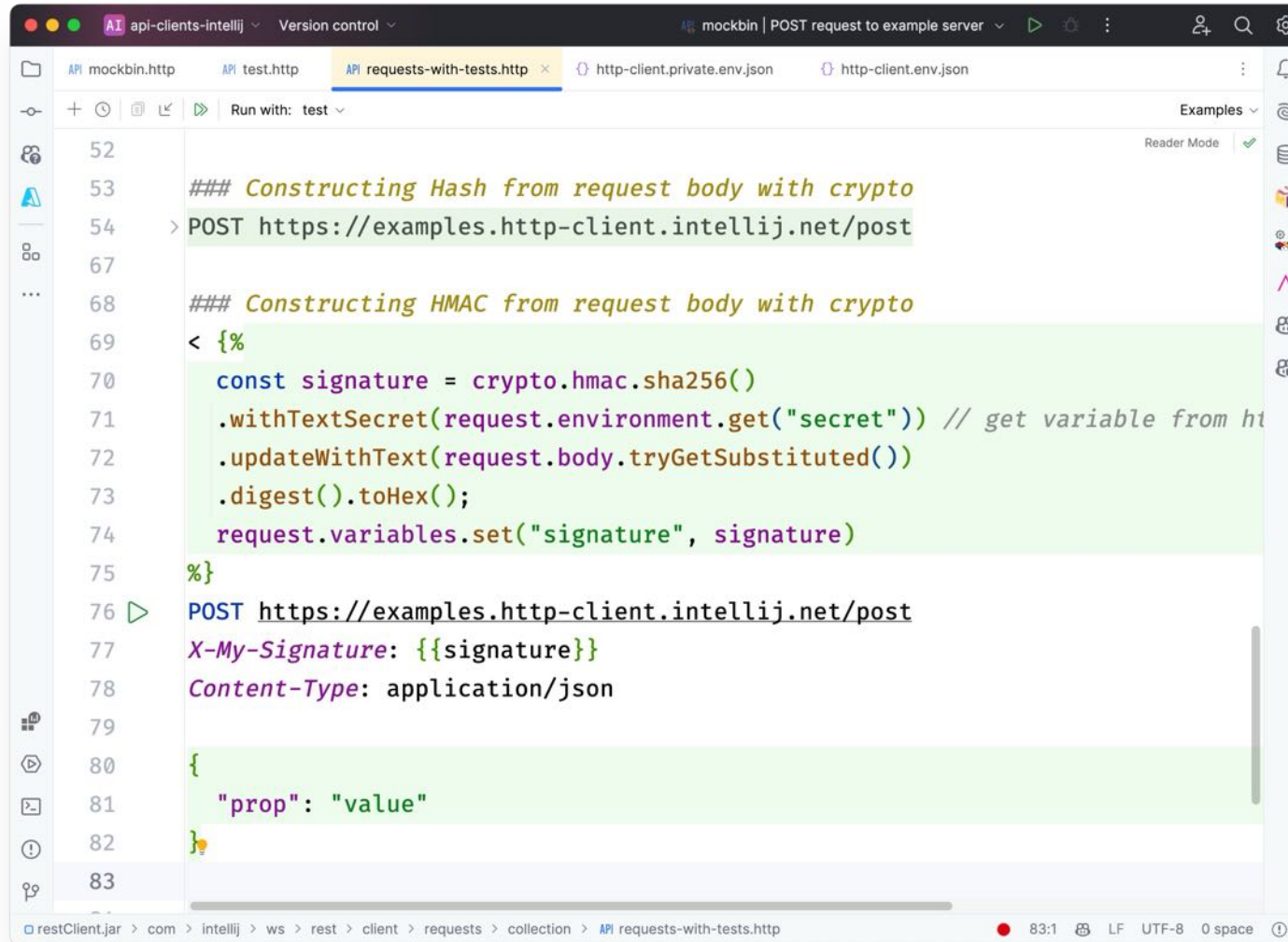


IntelliJ IDEA HTTP-Client: OAuth-Konfiguration



```
1 {
2   "dev": {
3     "foo": "bar",
4     "Security": {
5       "Auth": {
6         "microsoft365": {
7           "Type": "OAuth2",
8           "Grant Type": "Authorization Code",
9           "Client ID": "{{clientId}}",
10          "Client Secret": "{{clientSecret}}",
11          "Token URL": "https://login.microsoftonline.com/{{appId}}/oauth2/token",
12          "Auth URL": "https://login.microsoftonline.com/{{appId}}/oauth2/authorize",
13          "Redirect URL": "https://app.example.net/account",
14          "Client Credentials": "basic",
15          "Scope": "openid"
16        }
17      }
18    }
19  }
20 }
21
```


IntelliJ IDEA HTTP-Client: Pre-Request Scripts



```
52
53 ### Constructing Hash from request body with crypto
54 > POST https://examples.http-client.intelliij.net/post
55
56
57
58 ### Constructing HMAC from request body with crypto
59 < {%
60   const signature = crypto.hmac.sha256()
61     .withTextSecret(request.environment.get("secret")) // get variable from ht
62     .updateWithText(request.body.tryGetSubstituted())
63     .digest().toHex();
64   request.variables.set("signature", signature)
65 }%
66
67 > POST https://examples.http-client.intelliij.net/post
68 X-My-Signature: {{signature}}
69 Content-Type: application/json
70
71 {
72   "prop": "value"
73 }
74
75
```

IntelliJ IDEA HTTP-Client: Testing

The screenshot displays the IntelliJ IDEA HTTP-Client interface. The main editor shows a test script with the following content:

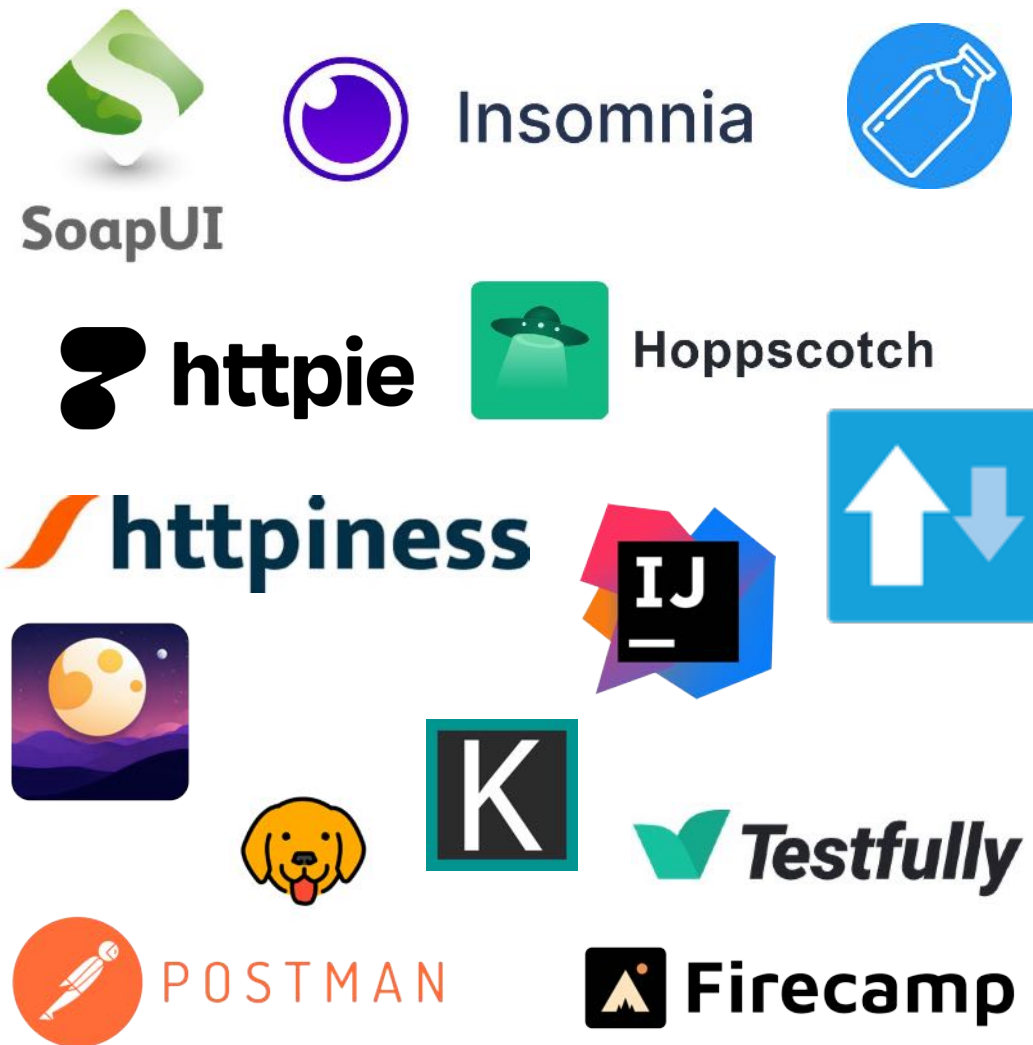
```
1 ### Successful test: check response status is 200  
2 GET https://examples.http-client.intellij.net/status/200  
3  
4 > {%  
5   client.test("Request executed successfully", function () {  
6     client.assert(response.status === 200, "Response status is not 200");  
7   });  
8 %}
```

Below the editor, the 'Tests' tab is active, showing the execution results:

- ✓ Tests passed: 1 of 1 test – 1 ms
- ✓ Test Results 1 ms
 - ✓ Tests 1 ms
 - ✓ Request executed successfully 1 ms

Fazit

Wofür haben wir uns entschieden?



Vielen Dank! Fragen?

andreas.siegel@pentacor.de

[linkedin.com/in/andreassiegel](https://www.linkedin.com/in/andreassiegel)

