

Dev Proxy

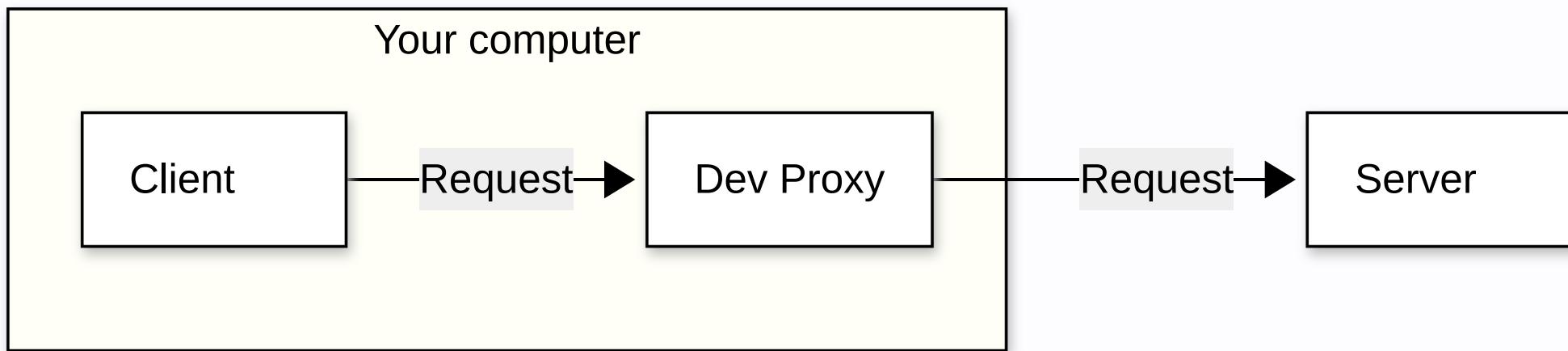
Dev Proxy is a tool that helps you effortlessly test your app beyond the happy path.

Story

September 2025 CloudFlare outage on their Cloudflare Dashboard due to repeated calls to their own API - the cause was a `useEffect` without any dependencies.
DDoS themselves.

What is Dev Proxy?

Dev Proxy is a forward proxy that you can use to intercept and modify requests from your application to any target server. With Dev Proxy, you can:



For and against (Pro et contra)

Pros

- You can mock whatever backend scenario you want.
- Use whatever script or language you want.
- It was like it was never there - bypass the proxy to return to your regular behaviour.

Cons

- Takes time to setup.
- Another dependency to install.
- Small learning curve.

Use Cases

- **Simulate API errors** - on internal (ie. localhost) and external (ie. Microsoft Graph).
- **Add latency** to requests to so you can test your UI display loading data messages.
- Fire **rate limits** and **handle throttling** - test with the `Retry-After` header.

- Stand up **mock APIs and data** - for when the backend is not ready yet
- **Intercept OpenAI-compatible requests** to **analyze costs**
- Check if your API is making requests with **least permissions** with API Center

Get Started

...with Dev Proxy
from the Dev Proxy
Team



Local Development

-  Dev Proxy will set itself as the system proxy. You can run commands `Invoke-WebRequest` or `curl`.
-  Chromium* based browsers (Edge/Chrome) bypass system proxy settings for localhost URLs - you need to exclude localhost URLs from the bypass list.

*See [documentation](#) for Firefox.

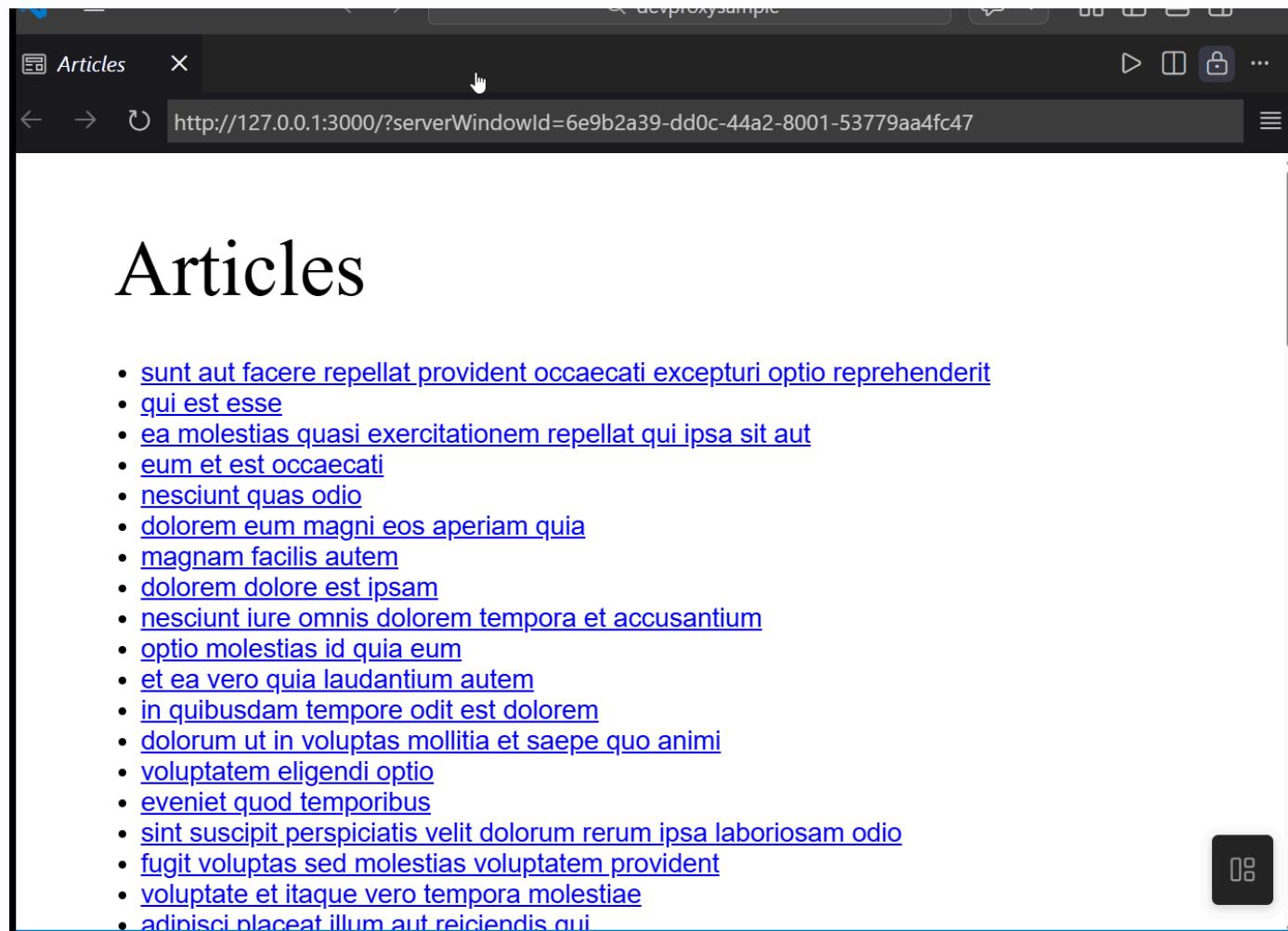
How to run locally

Run this to test local API's

```
taskkill /f /im msedge.exe  
cd "C:\Program Files (x86)\Microsoft\Edge\Application"  
. ./msedge  
--proxy-bypass-list=<-loopback>  
--proxy-server="127.0.0.1:8000"
```

⚠ Save your work beforehand ⚠

Usage



Dev Proxy Toolkit

[VSCode extension](#) is a
wrapper to the CLI.



Config

Open the VSCode command palette **CTRL+SHIFT+P**
and 'Dev Proxy Toolkit: Create configuration file'

Creates **.devproxy/devproxyrc.json**

```
{  
  "$schema": "https://raw.githubusercontent.com/dotnet/dev-proxy/main/schemas/v1.0.0/rc.schema.json",  
  "plugins": [  
    {  
      "name": "RetryAfterPlugin",  
      "enabled": true,  
      "pluginPath": "~appFolder/plugins/DevProxy.Plugins.dll"  
    },  
    {  
      "name": "GenericRandomErrorPlugin",  
      "enabled": true,  
      "pluginPath": "~appFolder/plugins/DevProxy.Plugins.dll",  
      "configSection": "genericRandomErrorPlugin"  
    }  
,  
  "urlsToWatch": [  
    "https://jsonplaceholder.typicode.com/*"  
  ],  
  "genericRandomErrorPlugin": {  
    "errorsFile": "devproxy-errors.json"  
  },  
  "rate": 50,  
  "logLevel": "information",  
  "newVersionNotification": "stable",  
  "showSkipMessages": true,  
  "showTimestamps": true  
}
```

Plugins

Use [plugins](#) or make your own

- `Auth` - Simulates authentication and authorization using API keys or OAuth2.
- `CachingGuidancePlugin` - Shows a warning when Dev Proxy intercepted the same request within the specified period of time.
- `CrudApiPlugin` - Simulates a CRUD API with an in-memory data store.

- `GenericRandomErrorPlugin` - Fails requests with a random selected error from file containing mocked errors.
- `LatencyPlugin` - Delays responses by a random number of milliseconds from the configured range.

- `EntraMockResponsePlugin` - Mocking auth flow API requests.
- `ExecutionSummaryPlugin` - Creates a summary of the requests that pass through the proxy.
- `MockGeneratorPlugin` - Generate mocks from the request.

- `MockRequestPlugin` - test webhooks in your client from dev proxy.
- `RateLimitingPlugin` - Simulates rate-limit behaviors.
- `RewritePlugin` - use rewrite rules - test example.com vs example.local
- `UrlDiscoveryPlugin` - creates a list of requested URLs.
- ... and much more (including your own)



Mock and CRUD Plugin

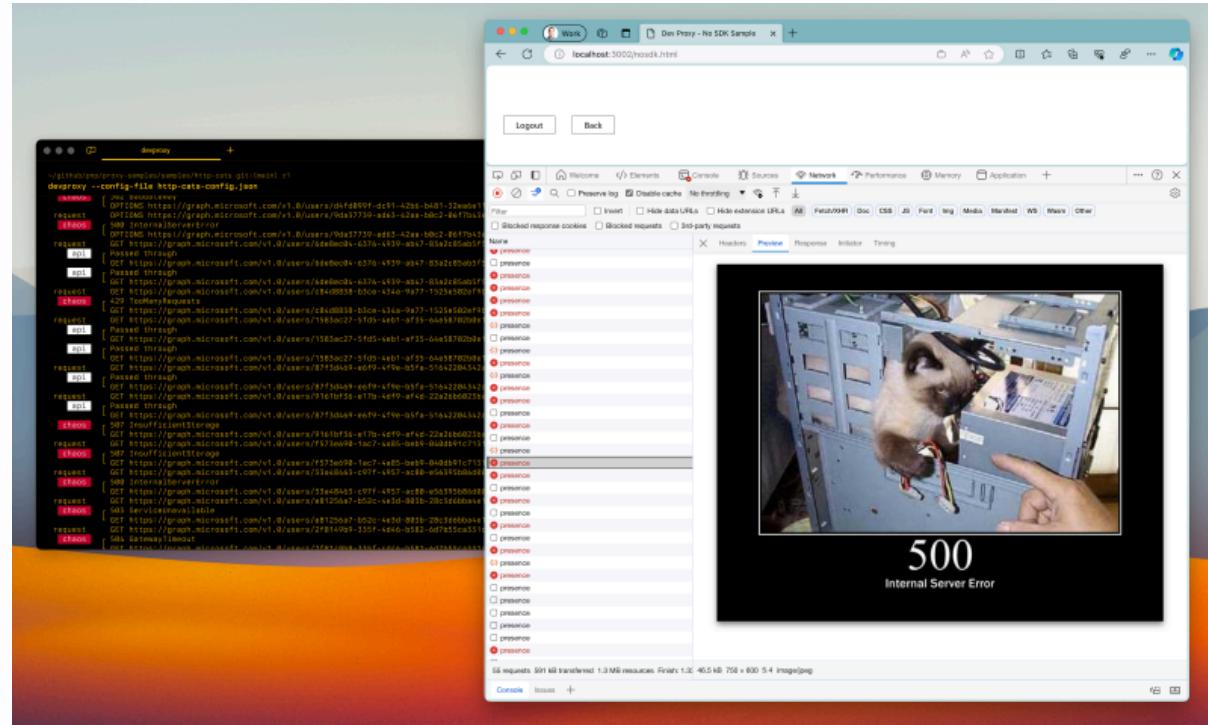
Return fixed data when your backend is not ready

- Different data on `nth` request
- Can deliver binary data
- Create a mock CRUD API - mutates in memory and resets on restart.
- Access via `devtunnel` over the internet - have your cloud apps connect to your machine.
- Supports Microsoft Entra

Samples

github.com/pnp/proxy-samples

- http cats - return cat images when simulating erroneous HTTP status codes for Microsoft Graph.



CloudFlare

What about the CloudFlare outage and how could Dev Proxy have helped in this situation? The plugin CachingGuidancePlugin could have been run on the developers machine and they would have seen that the API was getting called too frequently.

Resources

 View Online neave.dev/devproxy

 Download [PowerPoint](#) and [PDF](#)



Contribute on  [GitHub](#)