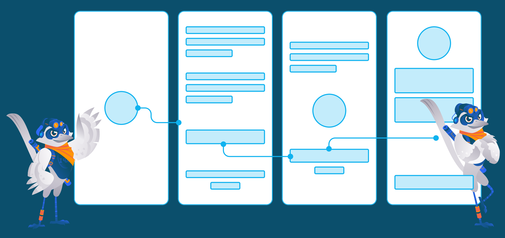
Parallel Development

Mock Internal APIs and Microservices

Prevent bottlenecks in your development process by parallelizing development work between API consumers and API developers



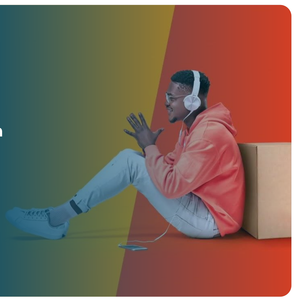
The Challenge

When different teams are dependent on unfinished APIs, development grinds (e.g., a front-end team waiting for a back-end API, or an internal team waiting for a partner API).

The Solution

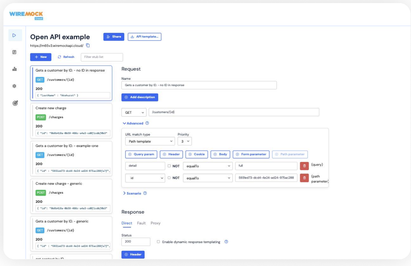
WireMock Cloud keeps your dev organization productive by allowing development to continue around realistic mock APIs that run as a service in your environment. Advanced features such as state management, proxying, and contract testing give you more flexibility to build and test your app in real-world conditions.

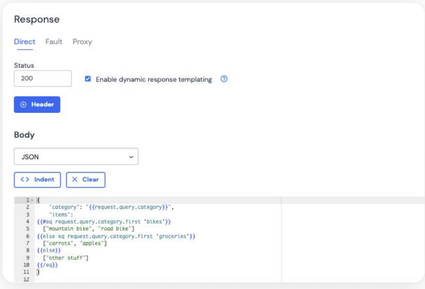
> **"We had a challenge of coordinating work between front-end and back-end teams. Using WireMock Cloud, we were able to parallelize the work. We could take 3 weeks off of every feature development by using this approach - That was a complete shift in our development efficiency"**
>
> **Joao Falcao**
> CTO, Jumia



Quickly spin up new mocks that run as a hosted service

- Build and manage your mocks via the web UI, and scale usage with the WireMock Cloud API
- Run WireMock as a hosted service in our Cloud, or on a Kubernetes cluster
- Import OpenAPI / Swagger specs to instantly scaffold mock APIs



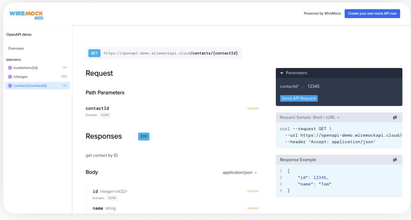


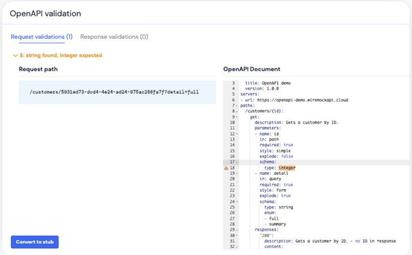
Remove dependencies and keep every team building and testing

- Continue developing API-dependant features with realistic mocks that simulate real-world condition
- Dynamically generating responses based on runtime data and context to test downstream behavior
- Keep developers on the same page and identify 'drift' from pre-agreed API definitions

Prototype, develop, and test APIs collaboratively to prevent production issues

- Test your system end-to-end against realistic mocks for smoother integration
- Collaborate on mocks to identify issues with API design or implementation early
- Validate system resilience by testing worst-case scenarios including delays throttling, and errors





Keep your APIs mocks trustworthy with continuous contract testing

- Validate your mock's requests and responses against the OpenAPI definition
- Keep your mock APIs reliable when the mocked API changes
- Keep developers on the same page and identify 'drift' from pre-agreed API definitions