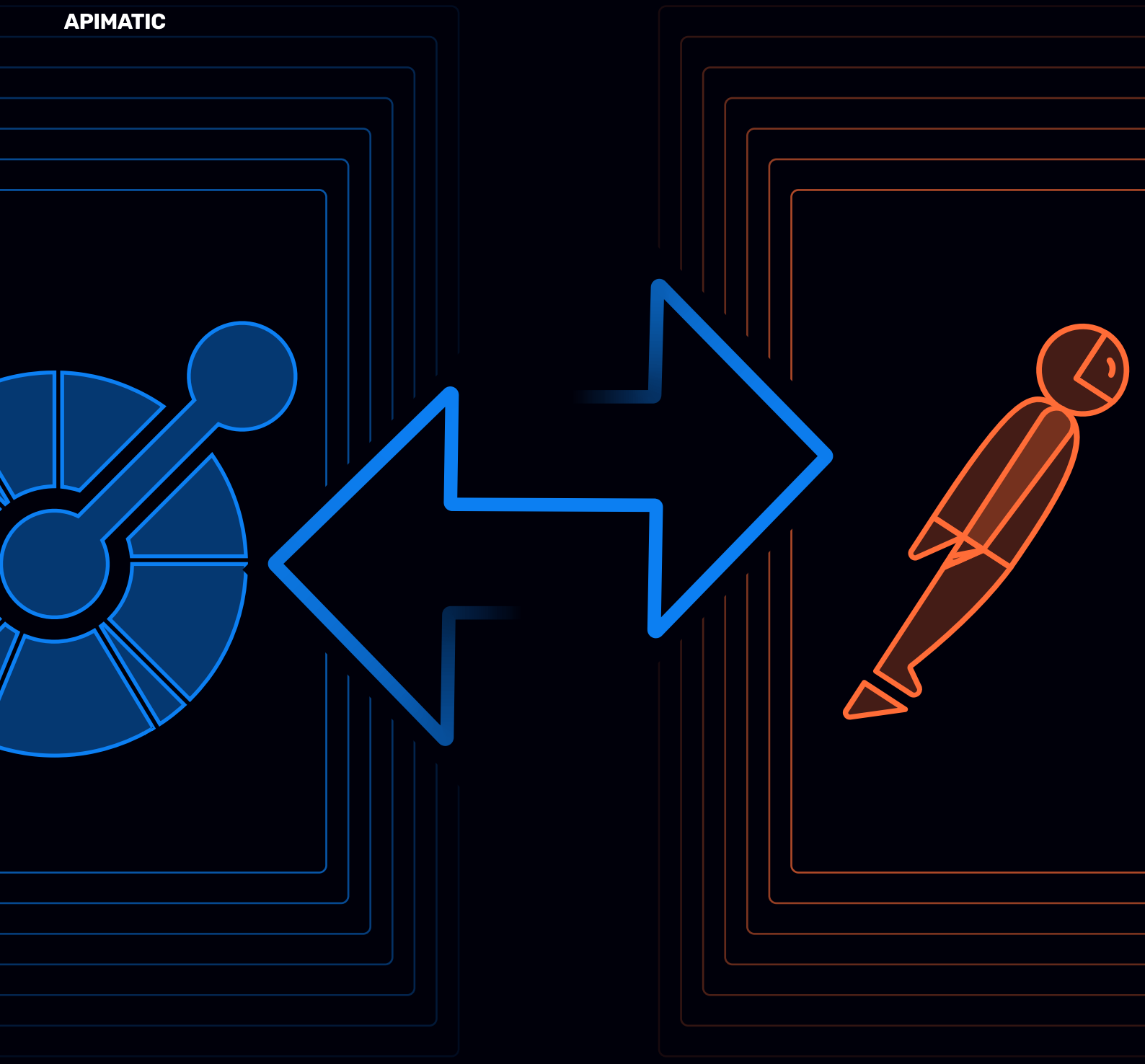
APIMATIC



CASE STUDY

One Utility Bill Auto-Converts OpenAPI Schemas to Postman Collections via APIMatic's Transformer API

Industry
Energy, Billing, Software

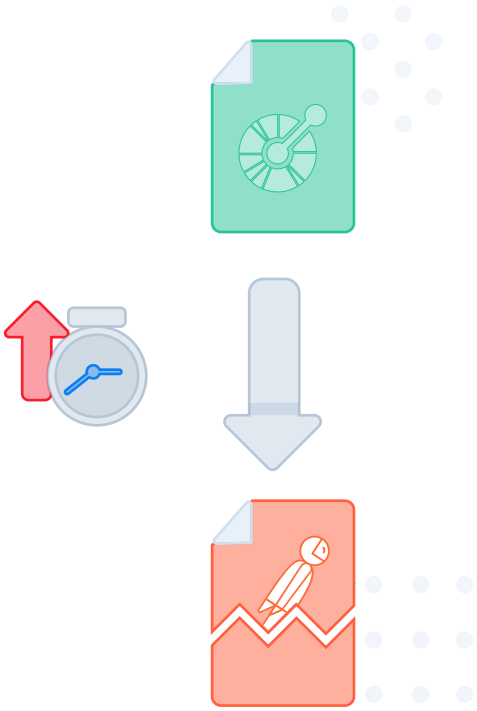
Headquarters
Newcastle upon Tyne, United Kingdom

oneutilitybill.co

One Utility Bill is a bill bundling platform that makes utilities as easy as they should be. They consolidate utilities, broadband, and media subscriptions into one easy monthly payment. This makes things more economical, quick and simple for tenants, landlords, and agents. Bills are easier to handle and property management is more streamlined.

One Utility Bill provides some of its services via the Notify platform. Notify informs utility providers of tenancy changes in the United Kingdom, ensuring that billing periods are correctly managed. For this purpose, the Notify API contains the tools needed to programmatically submit and manage data to the Notify platform.

The Pain Points: Incomplete Conversion from OpenAPI to Postman Collections



As businesses become increasingly API-centric, One Utility Bill relies on digitizing its services via APIs. The Notify API is hosted for users publicly using Postman, however, the API is an OpenAPI 3.0 schema that needs to be converted into a Postman Collection to be hosted.

The Notify API follows the JSON:API specification which is a well-defined, heavily nested API format. Using Postman's OpenAPI to Postman converter converted the API definition, but it didn't retain the structure or examples and OpenAPI inheritance essential to One Utility Bill.

This increased the post-processing time as the developers had to manually spend time on fixing the incomplete features in the API definition, along with an additional task every time the OpenAPI schema was updated.

Challenges

- Incomplete conversion to Postman
- Additional time overhead
- Manual conversion on each OpenAPI schema update

Automating OpenAPI to Postman Collection Conversion with APIMatic's Transformer API

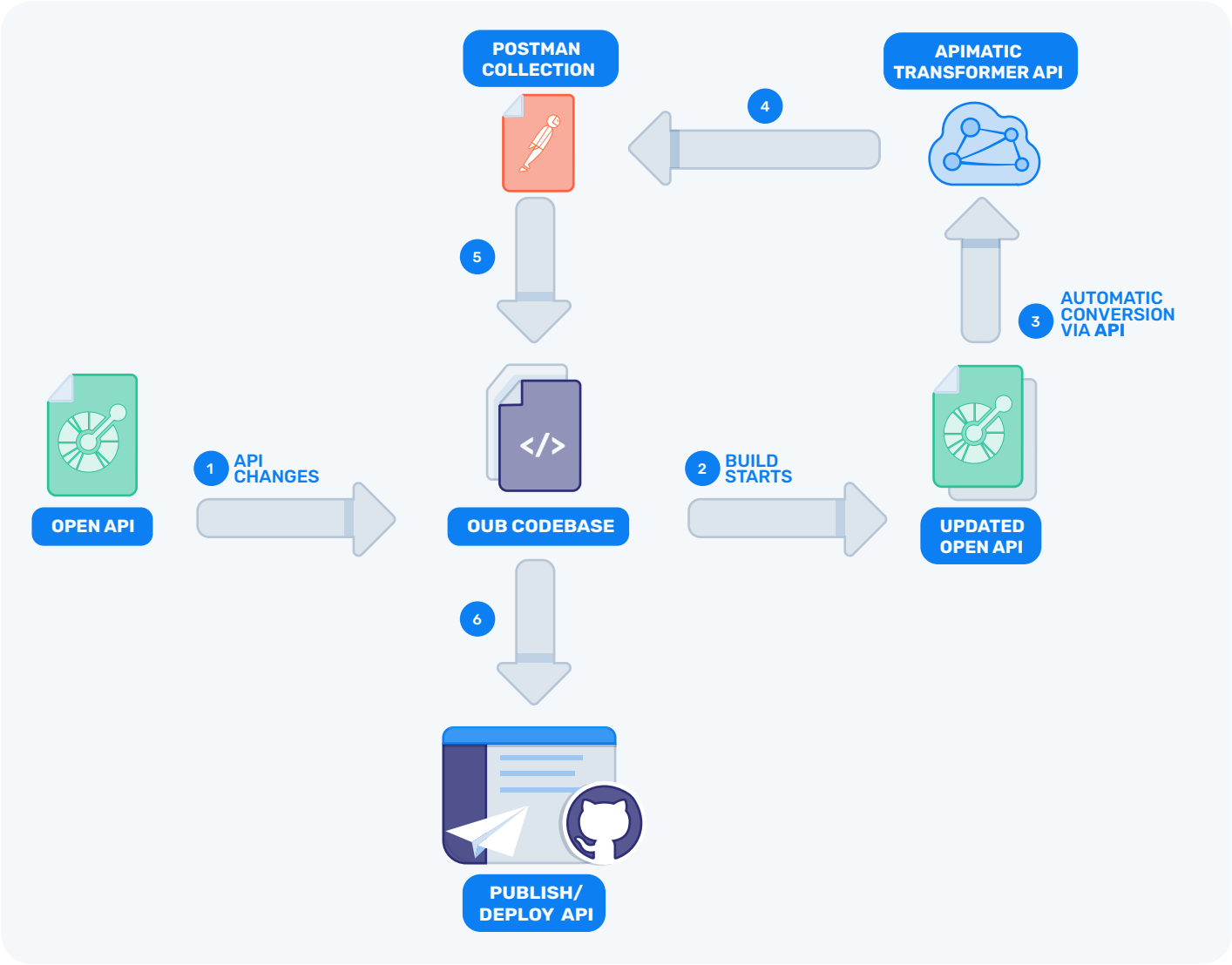
One Utility Bill needed to increase their flexibility in terms of tools used for their API, and use an OpenAPI schema within Postman. They discovered APIMatic's Transformer, which converts API definitions to and from 10+ specification formats, including OpenAPI, Postman, RAML, and more.

"APIMatic's Transformer processed the OpenAPI schema much better than Postman's tooling; the Postman Collection produced was as we expected with all the OpenAPI inheritance and composition followed correctly, with all of the examples we had defined in the OpenAPI schema." - Joe Keilty, Senior DevOps Engineer

The API Transformer can be used via Web or can be integrated into CI/CD pipelines via the Transformer API that validates and transforms any given API definition.

One Utility Bill's developers define the OpenAPI schema in their codebase. They maintain a CI/CD job that automatically runs during deployments if it is detected that their OpenAPI schema has changed. This executes a script that converts the OpenAPI schema to a Postman Collection v2 format via APIMatic's Transformer API. The resulting Postman Collection file is then used to update their publicly accessible documentation.

"This has turned what may have been a manual process that a developer would have to remember to do each time into something which works entirely in the background whenever we make changes to our API."



The Impact: Automatic and Accurate Conversions without Human Intervention

Impact

Accurate conversion from OpenAPI to Postman

Reduced time overhead, from 30 minutes to seconds

Automatic conversion on each OpenAPI schema update

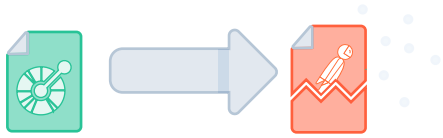
One Utility Bill has not hired any developer or technical writer to maintain and convert their API definitions since APIMatic was added to their landscape. It also accounts for one less thing that the developers need to remember for each deployment, which lets them focus on the actual changes they have released.

"We would not have been able to host our API documentation in the way we have done if it wasn't for the APIMatic Transformer."

"The availability of the APIMatic Transformer API allowed us to take what would have been a 15 - 30 minute manual error-prone task every time we deployed API changes into something that happens automatically in CICD in less than a minute." - Joe Keilty, Senior DevOps Engineer

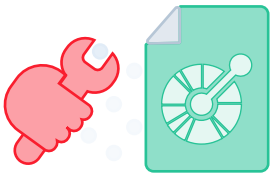
Before

Incomplete conversion to Postman



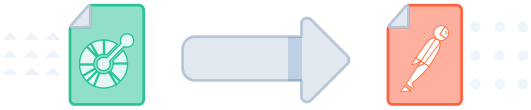
Additional time overhead

Manual conversion on each OpenAPI schema update

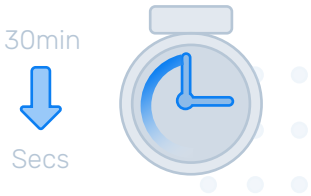


After

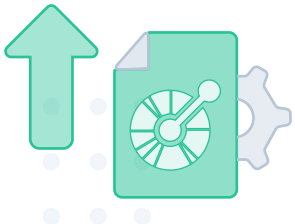
Accurate conversion from OpenAPI to Postman



Reduced time overhead, from 30 minutes to seconds



Automatic conversion on each OpenAPI schema update



Level 7, 3 Albert Street,
Auckland 1010, New Zealand

www.apimatic.io
APIMatic © 2021, All Rights Reserved.