

# Postman

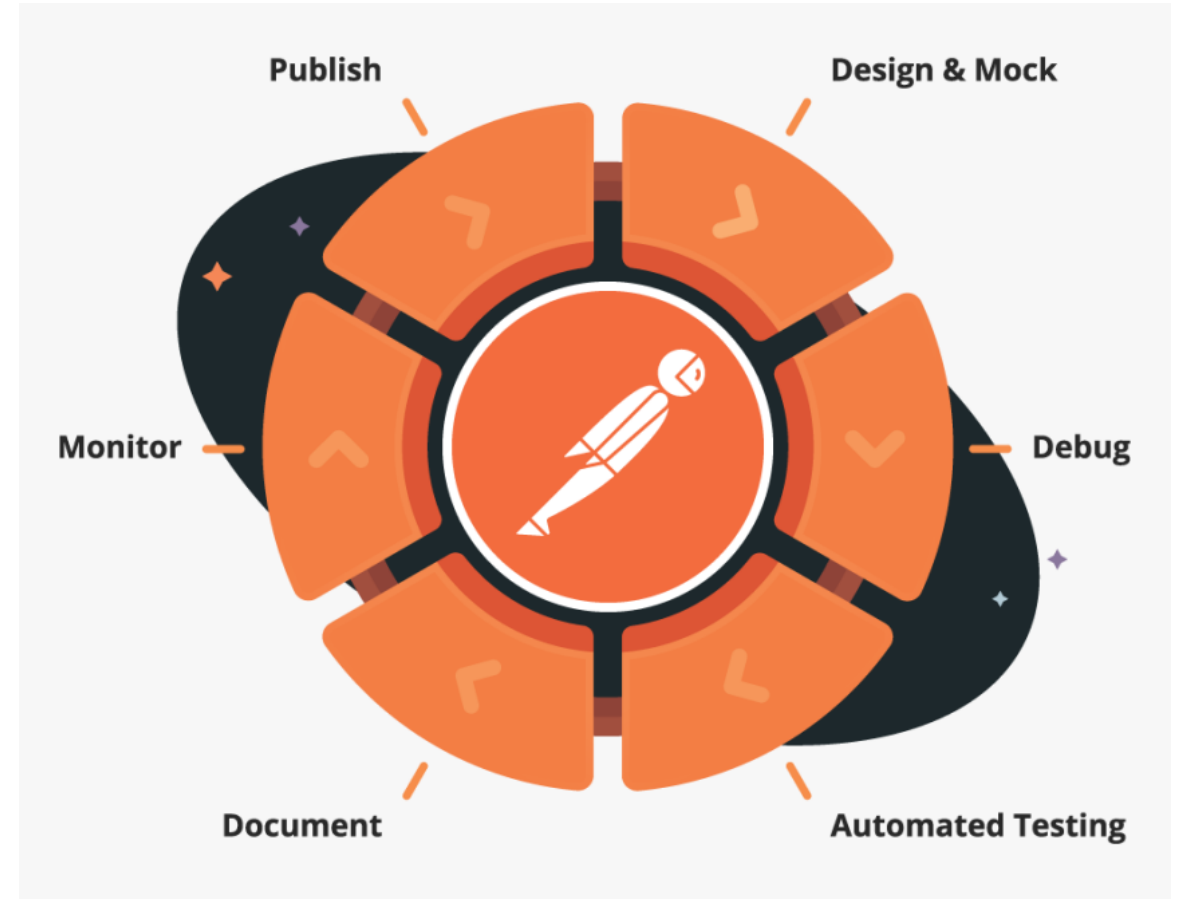


Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration."



# Features

- Make Request(s)
- Test
- Publish, Share & Collaborate with your Team or the Public
- Documentation
- Monitor your API Status
- Mock not yet created APIs based on API Definitions



# Create Requests

Build collections of requests that read, create, delete data in an API no matter if it's a GraphQL, REST or even gRPC API.

- GET/POST/PUT/DELETE/HEAD
- Query Parameters
- set Headers (including Cookies)
- Form, JSON Bodies
- upload files
- organize those requests in nested folders
- add description to parameters and variables for easy re-use

# Flexibility

Create a collection once, use in different environments, for different user(s) and use-cases.

- variables on different levels:
  - collections
  - environment(s)
  - global
- re-use the variables anywhere ( `{{variableName}}` )

More: [Defining Variables](#)

# Random Variables

Postman uses Faker to provide a whole bunch of different randomly created variables that can be used especially in create request.

Think of:

```
{  
  "username": "{{randomFirstName}}-{{randomLastName}}",  
  "password": "{{randomPassword}}"  
}
```

Creating a new randomly created user each time and uses this user for subsequent requests.

More about Dynamic Variables

# Tests

Each requests runs Javascript Code when it's done in order to be able to test the results. This makes it easy to add simple (and even complex) assumptions to a single request or even folders or whole collections.

Some Examples:

- valide response bodies agains certain schema(s)
- check for presence or values of header(s)
- re-use response data for later request(s)

More Test & Scripting: <https://learning.postman.com/docs/writing-scripts/script-references/test-examples/>

Library API Reference

Library API Reference

+    ⋮

No Environment

⌵

🔗

Library API Reference

No Environment, just now

View Summary

Run Again

New

Export Results

All Tests

Passed (2)

Failed (0)

Iteration 1

GET

books

{{baseUrl}}/books

/ books

200 OK

185 ms

9.701 KB

Pass

Status code is 200

GET

book

{{baseUrl}}/books/{{id}}

/ book

404 Not Found

95 ms

281 B

This request does not have any tests.

POST

add book

{{baseUrl}}/books

/ add book

201 Created

117 ms

233 B

Pass

Status code is 200

PATCH

update book

{{baseUrl}}/books/{{id}}

/ update book

404 Not Found

103 ms

281 B

This request does not have any tests.

DELETE

delete book

{{baseUrl}}/books/{{id}}

/ delete book

404 Not Found

96 ms

281 B

This request does not have any tests.

1

# Run Collection

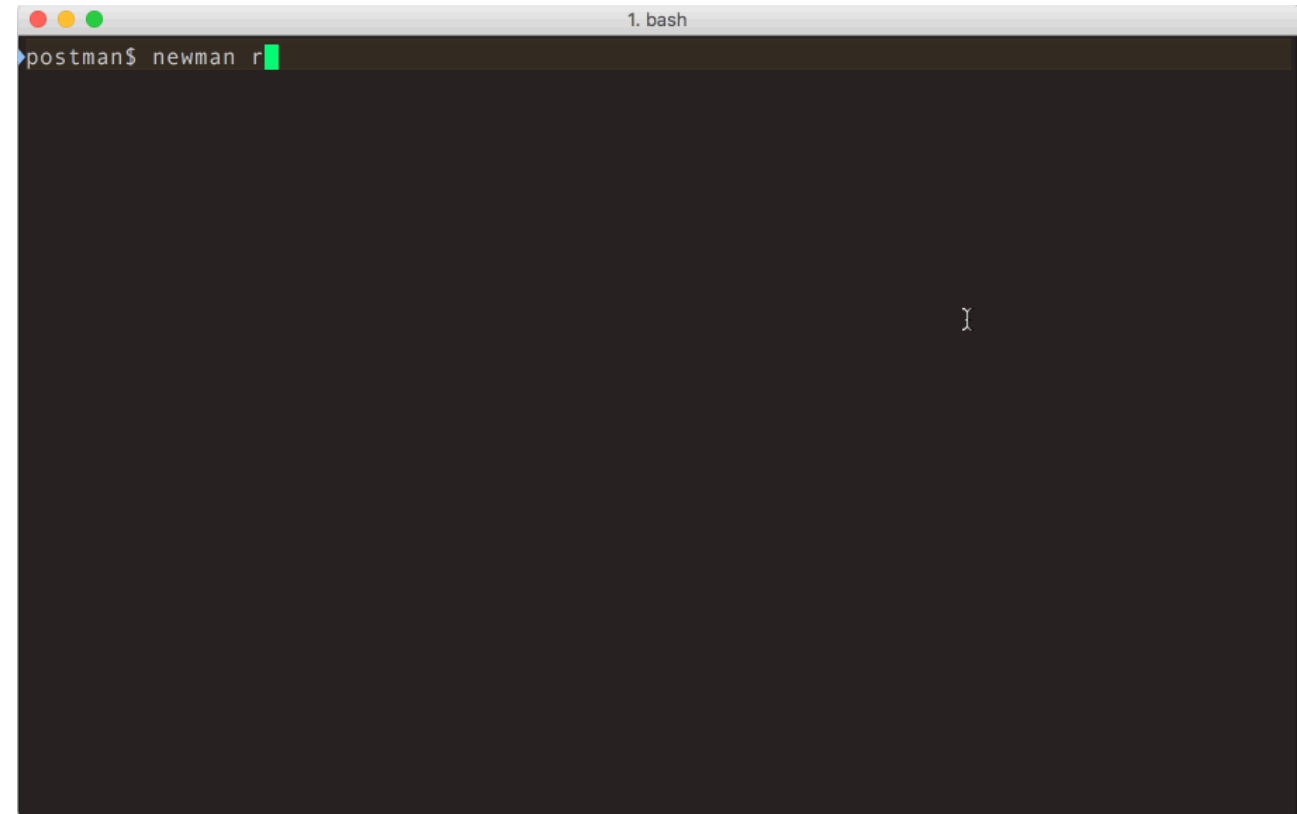
A collection can be run  
alltogether and all requests and  
tests are executed sequentially  
and result in a report:

# Automation

newman is a command line tool written in nodejs which can run collections and execute it's tests:

```
npx newman run <collection-json-filename>
```

More about [newman](#)





# Demo Time



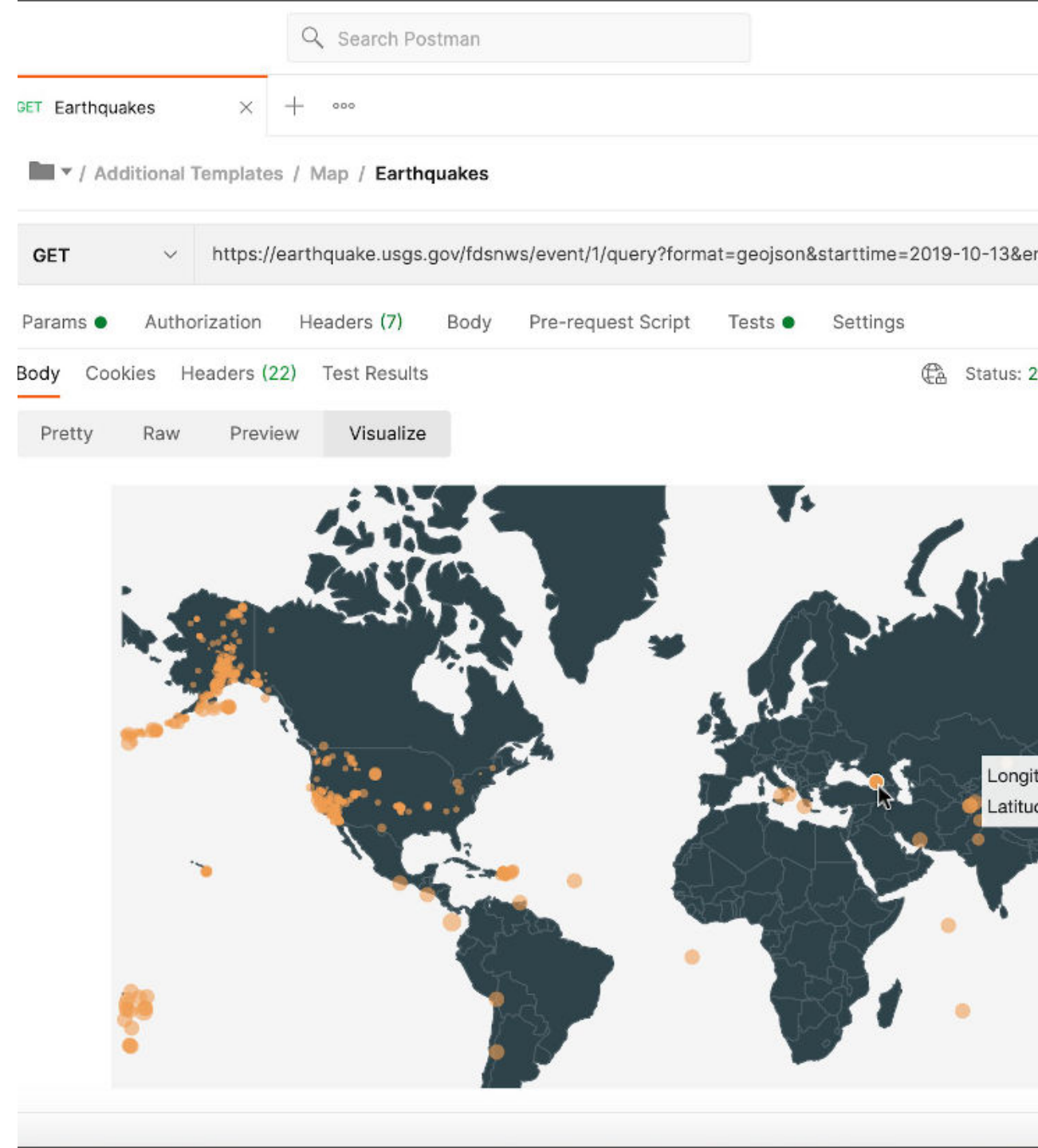
- Show simple GET request(s)
- Show GraphQL Request
- Show Test
- Show Collection Run
- Show NewMan
- Show Visualization
- Show Monitoring

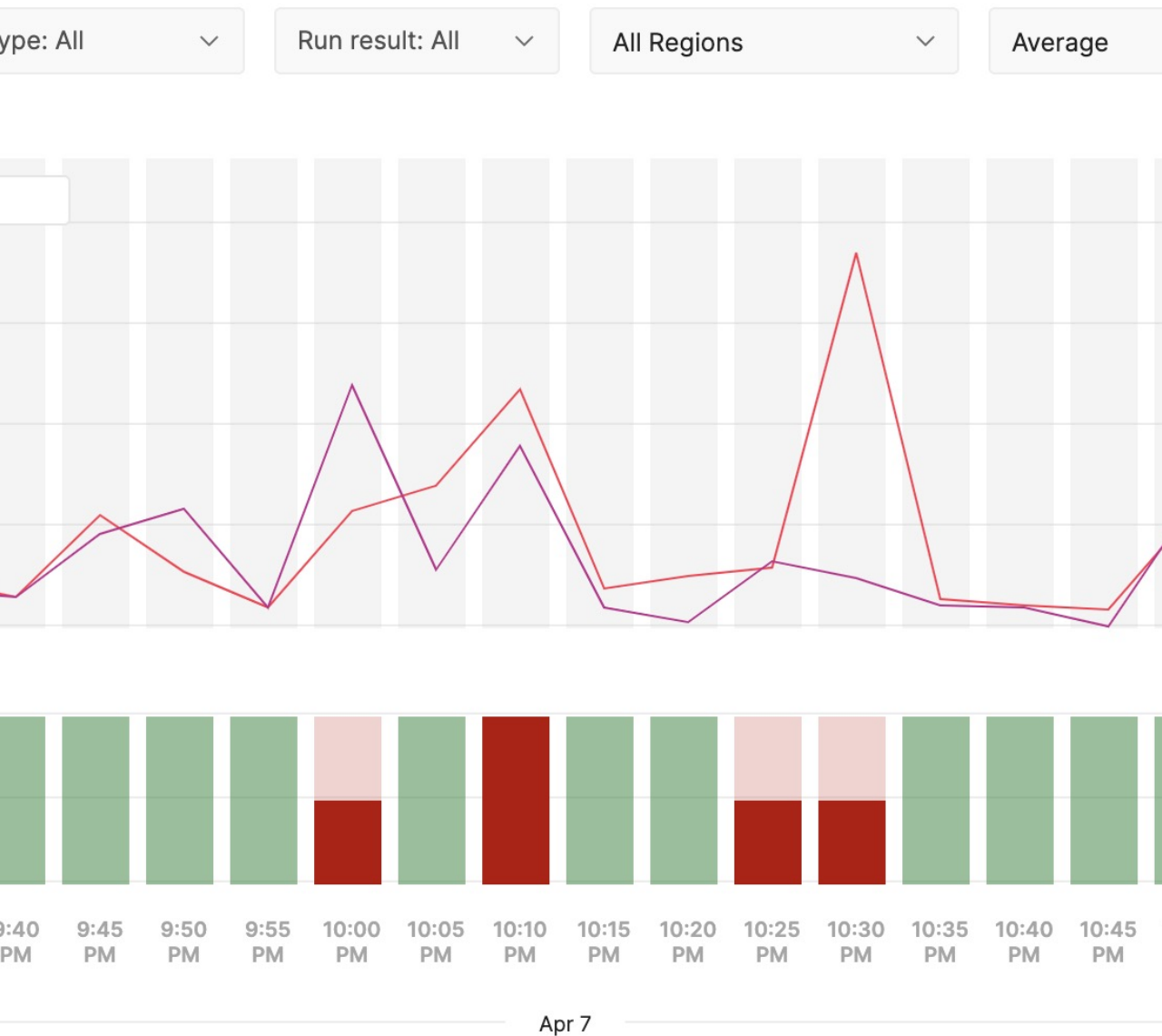
# Outlook

# Visualization

Show data from an API in (interactive) visualisations like tables, charts, maps etc.

More about [Visualization](#)





# Monitoring

Run collections every n-minute/hour and record the results of it

More about [Monitoring](#).

# Others

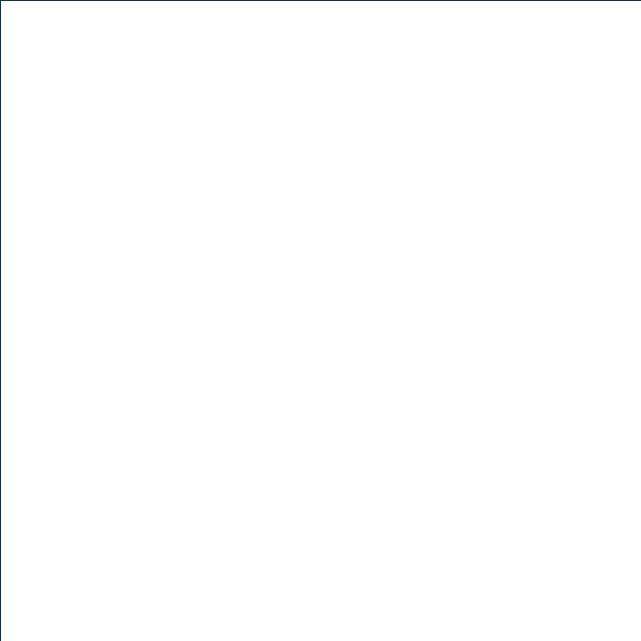
- Sharing
  - Share API Definitions, Documentations, Monitors with the public
  - checkout [Bitly API on Postman Explorer](#)
- Mocking
  - Create API definitions and start using them without having a server
- Flows
  - Visually program and connect multiple API requests and process results

Try It! It's free: <http://postman.com>

# Resources



- [Postman API testing by example](#)
- [Official Documentation](#)
- [Automation tricks with newman](#)



**Thanks for  
listening!**