



Automated API Testing

Postman & Newman

Hello!

I Am DANIEL HERRERO

You can contact me at
daniel.herrero@kaleidos.net



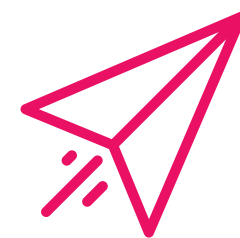


Postman



What it is

Postman
API development environment
Local desktop app / Web
Team synced remotely
Postman, Pro, Enterprise



Sample API

Heroes objs (Domain Classes)
Heroes ops (UrlMappings)
Grails



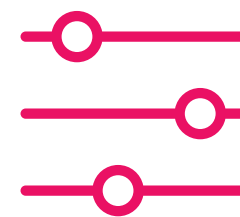
Concepts

Workspace, Collections,
Environments & Variables, Test
Scripting, Syncing, Running a
Collection, Documenting API,
Mocking API responses



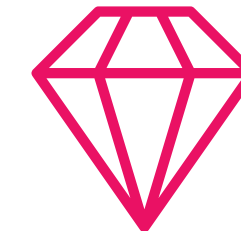
What it allows

Design & Mock
Debug & Testing
Automated Testing
Documentation
Monitoring
Postman API



Direct usage

Share collection.json on gitlab
For back-end & front-end
API Documentation from CLI
Run API Tests from CLI



What if

Begin With the End in Mind
CI/CD API Documentation
CI/CD API Testing





Postman



What it is

API development environment

Local desktop app / Web

Team synced remotely

Postman, Pro, Enterprise

Details

- ❑ Postman came to life in 2012 as a Chrome extension built as a side project.
- ❑ Native application for Windows, Mac, and Linux
- ❑ Newman command line utility
- ❑ SaaS cloud service





Postman



What it is

API development environment

Local desktop app / Web

Team synced remotely

Postman, Pro, Enterprise

Details

- ❑ Register a User Account
- ❑ Install the client locally
<https://www.getpostman.com/apps>
- ❑ Acces the same content on the Web





Postman



What it is

API development environment

Local desktop app / Web

Team synced remotely

Postman, Pro, Enterprise

Details

- ❑ Every change in the client is instantly synced to the cloud
- ❑ Activity log record

Demo

- ❑ Postman client (First look)
- ❑ “View In web” / Manage Team
- ❑ Changes get logged





Postman

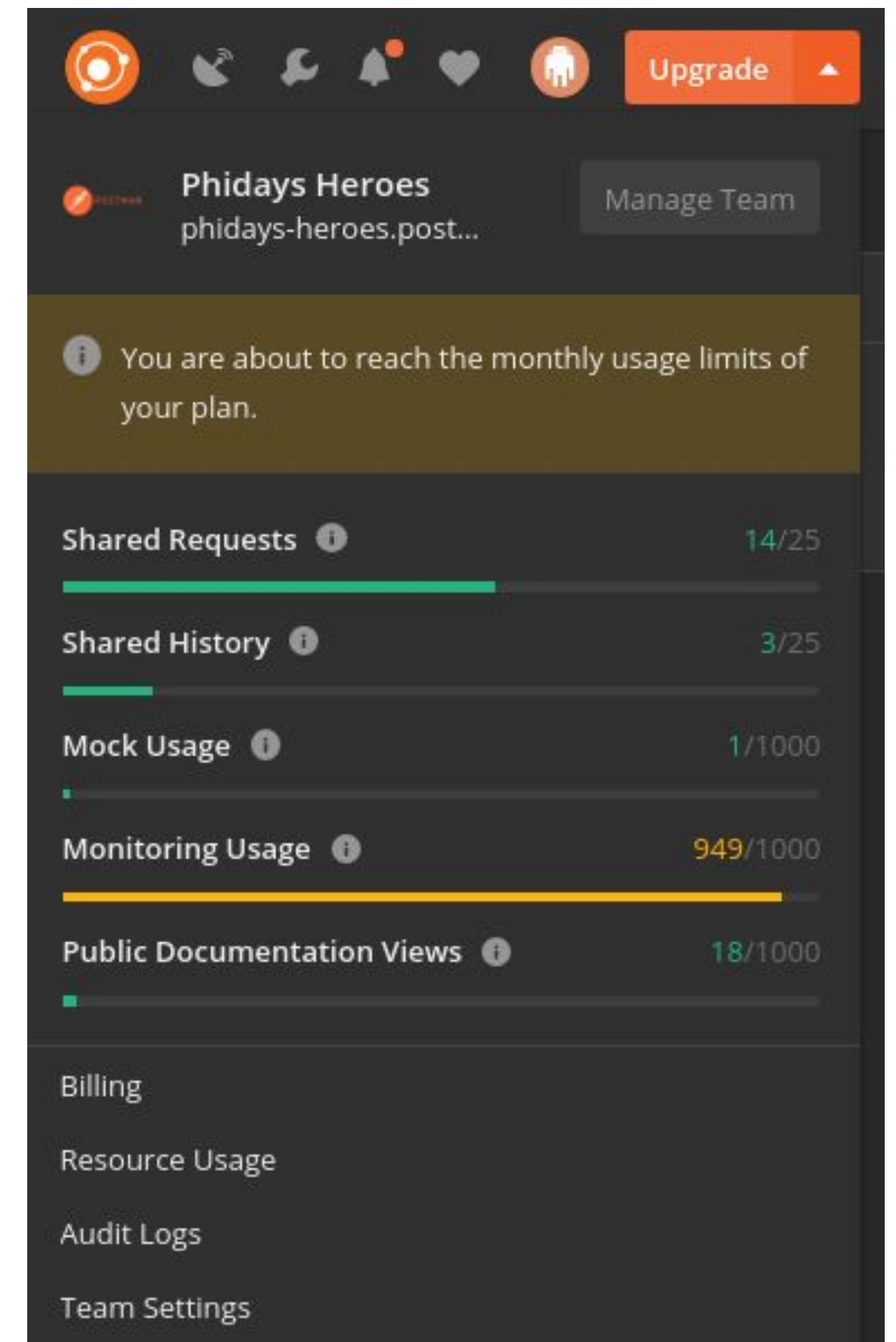


What it is

API development environment
Local desktop app / Web
Team synced remotely
Postman, Pro, Enterprise

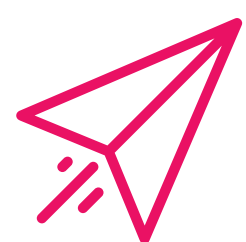
Details

- ❑ Usage limits: monitoring, mocking, documentation visits, ...
- ❑ Postman, Free (x5), 1000 calls/month
- ❑ Pro, 1000\$ (x10), 10K calls/month
- ❑ Enterprise 2000\$ (x10), 100K calls/month





Postman



Sample API

Heroes objs (Domain Classes)

Heroes ops (UrlMappings)

Grails

```
package phidays.heroes.ability

import groovy.transform.builder.Builder

@Builder
class Ability {
    String name
    String description
    Level level

    static constraints = {
        name maxSize: 255
        level enumType: 'string'
        description nullable: true
    }

    static mapping = {
        description type: 'text'
    }

    static enum Level {
        POOR( powerLevel: 0),
        TYPICAL( powerLevel: 1),
        GOOD( powerLevel: 2),
        EXCELLENT( powerLevel: 3),
        INCREDIBLE( powerLevel: 4),
        AMAZING( powerLevel: 5)

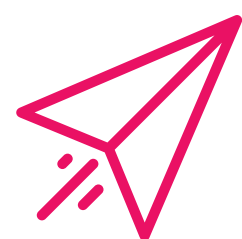
        private final int powerLevel
        Level(int powerLevel) { this.powerLevel = powerLevel }
    }
}
```

```
1 package phidays.heroes.character
2
3 import ...
4
5
6 @Builder
7 class Character {
8     String fullName
9     String alterEgo
10
11     static hasMany = [ abilities: Ability ]
12
13     static constraints = {
14         alterEgo maxSize: 255
15         fullName maxSize: 255, nullable: true
16     }
17
18     static mapping = {
19         id generator: 'increment'
20     }
21 }
22
```





Postman



Sample API

Heroes objs (Domain Classes)

Heroes ops (UrlMappings)

Grails

Details

- ❑ Endpoints definition
 - ❑ `"/abilities"(controller: 'ability', action: [GET: 'listAbilities', POST: 'createAbility'])`
 - ❑ `"/abilities/$abilityId"(controller: 'ability', action: [PUT: 'updateAbility'])`
 - ❑ `"/characters"(controller: 'character', action: [GET: 'listCharacters', POST: 'createCharacter'])`
 - ❑ `"/characters/$characterId"(controller: 'character', action: [PUT: 'updateCharacter', DELETE: 'deleteCharacter'])`





Postman



Concepts

Workspace, Collections, Environments & Variables, Test Scripting, Syncing, Running a Collection, Documenting API, Mocking API responses

Details

- ❑ Workspace = Control Center for Collaboration
 - Members & Permissions
 - Collections of requests
 - Environments
 - Mocks, Monitors
 - Documentation

- ❑ Collection = Group of requests you can organize into folders





Postman



Concepts

Workspace, Collections, Environments & Variables, Test Scripting, Syncing, Running a Collection, Documenting API, Mocking API responses

Details

- ❑ Environment = Group of variables that set the context for a collection of requests (Documentation, Running Tests, Mocking)
 - `{{hostURL}}` = <http://localhost:8080> "Heroes (localhost)"
 - `{{hostURL}}` = <https://49e89e74...> "Heroes (mocked)"
 - ❑ GET `{{hostURL}}/characters`
- ❑ Variables
 - Global variables: Its values are shared across environments
 - ❑ Example: Temporal tests variables (`{{abilityId}}`=1, `{{id}}`=1)
 - Environment variables: Unique variables that make sense in that context



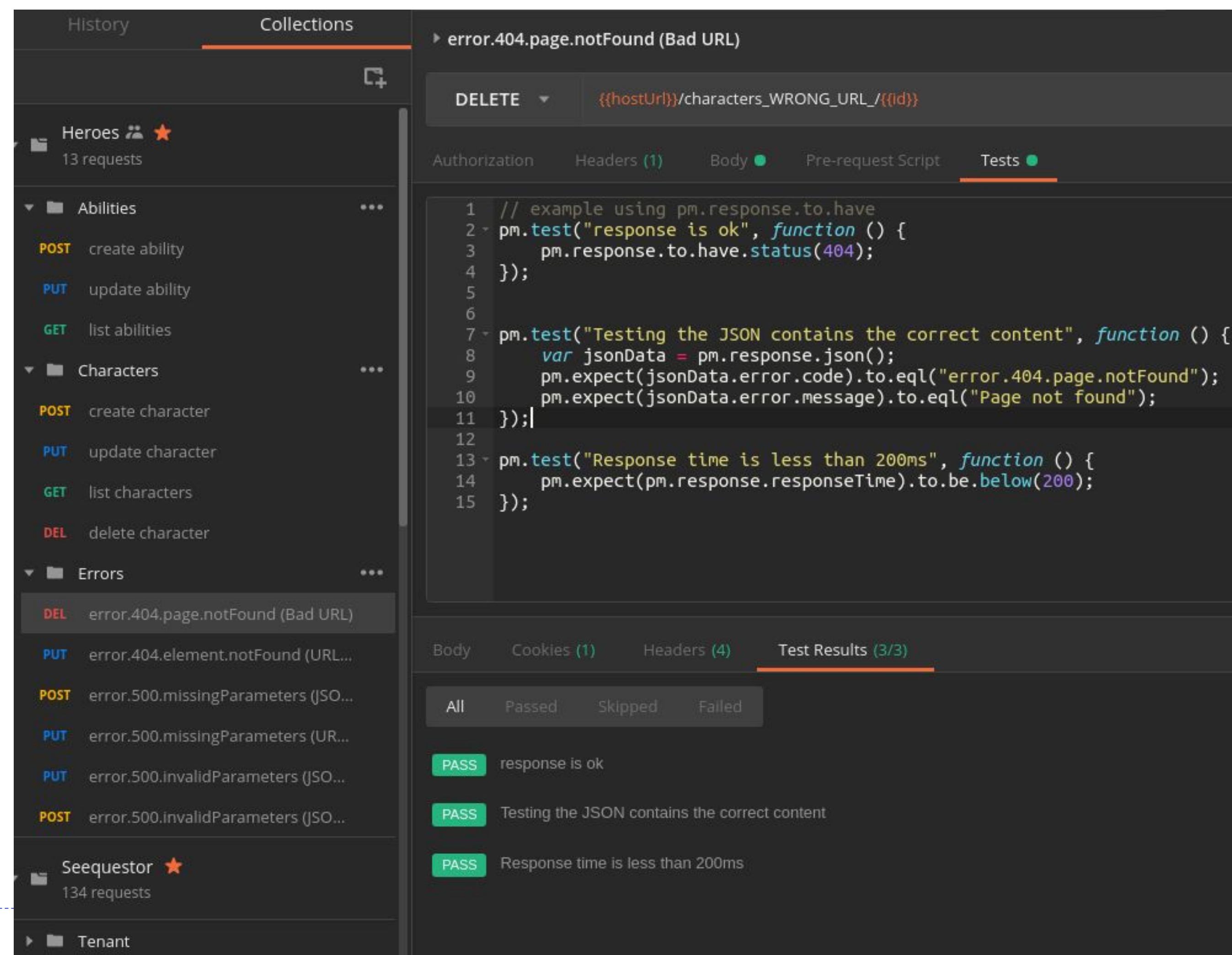


Postman



Concepts

Workspace, Collections, Environments
& Variables, Test Scripting, Syncing,
Running a Collection, Documenting
API, Mocking API responses





Postman



Concepts

Workspace, Collections, Environments & Variables, Test Scripting, Syncing, Running a Collection, Documenting API, Mocking API responses

Details

- ❑ Syncing = The ability to propagate local changes to the Postman cloud, and from that, to your team
- ❑ Running a Collection = To run the collection of requests sequentially, for an environment, validating their responses against the designed tests
- ❑ API Documentation = HTML automatically generated for a collection, generated from the requests, and any additional description written in Markdown
- ❑ API Mocking = Postman servers will reply requests according to your request specifications (working, or still simply documented requests)





Postman



What it allows

Design & Mock

Debug & Testing

Automated Testing

Documentation

Monitoring

Postman API

Details

1. Give the request a Name, a Description (Markdown), and Group it in folders
2. Define the Request (plenty of options & easy to use)
3. Define Example/Error Responses (+1), status codes (no endpoint available yet!)
4. Pretty render (json, xml, html, text)
5. Run Requests !!

Demo

- ❑ Postman client (Request Options)
- ❑ Drag & drop, contextual menus, shortcuts customizations





Postman



What it allows

Design & Mock

Debug & Testing

Automated Testing

Documentation

Monitoring

Postman API

Details

- ❑ Postman servers generate fake responses according to ...
 - All the collection requests (real)
 - All the example collection requests (documented)
 - For any defined environment
- ❑ Postman URL provided

- ❑ 1000 calls / month are too few for real usage





Postman



What it allows

Design & Mock

Debug & Testing

Automated Testing

Documentation

Monitoring

Postman API

Details

- ❑ Run individual requests
- ❑ Run collections (Collection Runner)
- ❑ Against any defined environment
 - HostDomain
 - Port
 - App URL
- ❑ Tests in Javascript

https://www.getpostman.com/docs/v6/postman/scripts/test_examples

Demo

- ❑ Postman client (Individual Requests)
- ❑ 3 example tests (POST error 500)
- ❑ Dynamic variables in consecutive tests (POST/UPDATE)





Postman



What it allows

Design & Mock

Debug & Testing

Automated Testing

Documentation

Monitoring

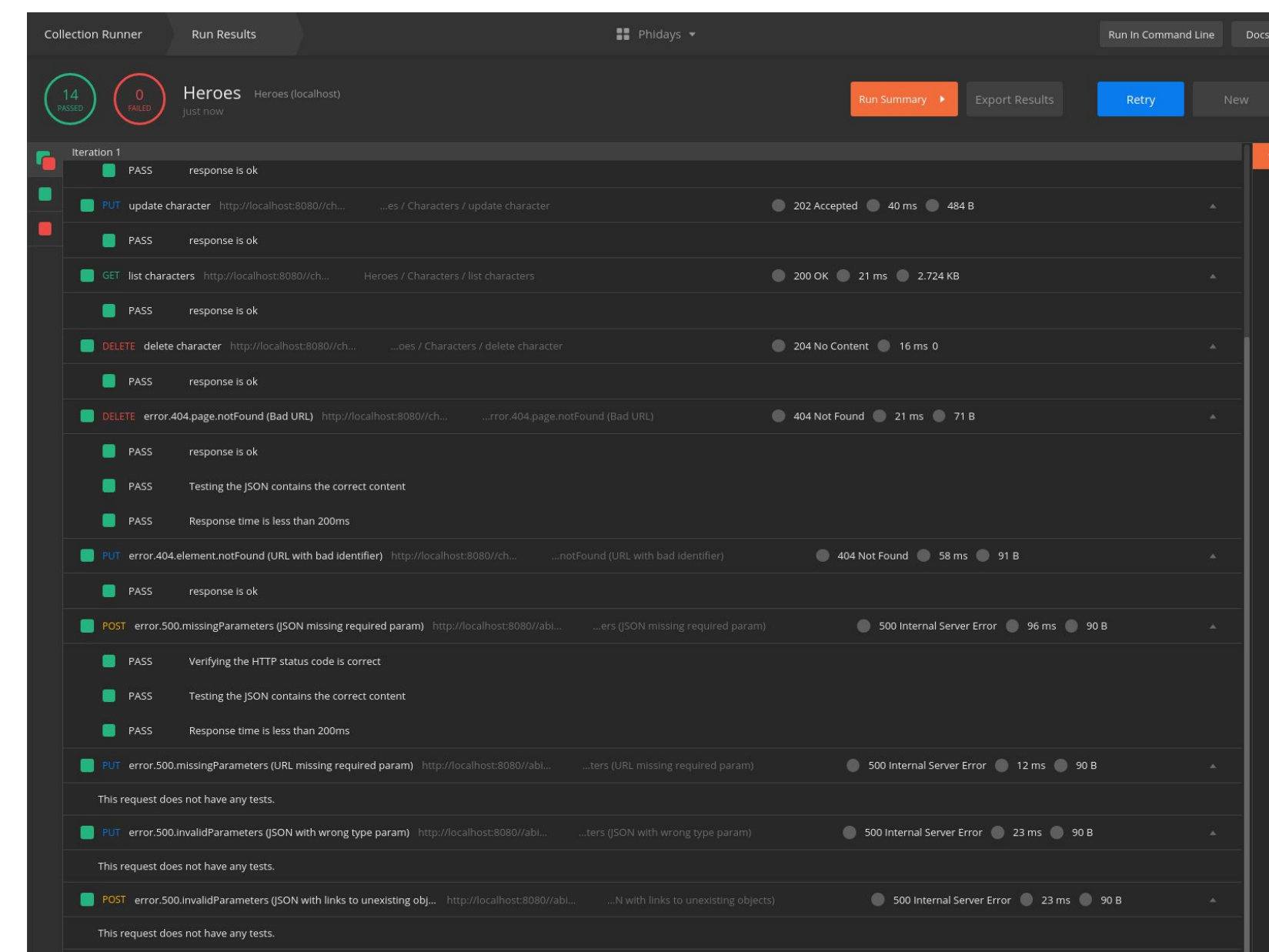
Postman API

Details

1. Choose an Environment
2. Choose a collection
3. Run from ...
 - Collection Runner (Client)
 - Web
 - CLI (Newman)
 - Monitoring (Postman cloud)

Demo

Collection Runner (Client)





Postman



What it allows

Design & Mock

Debug & Testing

Automated Testing

Documentation

Monitoring

Postman API

Details

1. Requests are directly documented
2. Add more API / Request details (Markdown)
3. Choose an Environment
4. Choose a collection
5. Publish documentation

Demo

- ☐ Publish collection (Client/Web)
- ☐ Access documentation





Postman



What it allows

Design & Mock

Debug & Testing

Automated Testing

Documentation

Monitoring

Postman API

Details

- ❑ API uptime, responsiveness, and correctness
- ❑ Reporting of monitoring results
- ❑ 1,000 free monitoring calls/month

Demo

- ❑ Postman Client (Create Monitoring)
- ❑ Collection > frequency > reports





Postman



What it allows

Design & Mock

Debug & Testing

Automated Testing

Documentation

Monitoring

Postman API

Details

- ❑ The Postman API allows you to programmatically access data stored in Postman account with ease.
- ❑ API key Access
- ❑ Example: <https://api.getpostman.com/collections>
- ❑ More info about [Postman API](#)





Postman



Direct usage

Share collection.json on gitlab

For back & front-end

API Documentation from CLI

Run API Tests (Newman) from CLI

Details

- ❑ A collection defines the requests
- ❑ An environment defines variables

⇒ Export it to “collection.json”

⇒ Export it to “environment.json”

We have everything we need !!

- ❑ Make it available
- ❑ Shared it with your team

⇒ Upload it to gitlab

⇒ Import json's in Postman clients

It allows to have an **updated API ...**

⇒ **Locally**

- Make any request
- Run the API tests
- Access API Documentation





Postman



Direct usage

Share collection.json on gitlab

For back & front-end

API Documentation from CLI

Run API Tests (Newman) from CLI

Details

- ❑ Sharing a collection allows **back-end** to ...
 - Run single requests: Make HTTP Requests to any endpoint
 - Run the collection: Assure no other endpoint is broken
 - Re-document the API (for new/updated requests)

- ❑ Sharing a collection allows **front-end** to ..
 - Review an updated API Documentation (always fresh-built from the requests)
 - Re-document the API (writing new requests proposals)

As there would only exist a unique collection ...

... Everybody would work over **the same API** !!





Postman



Direct usage

Share collection.json on gitlab

For back-end & front-end

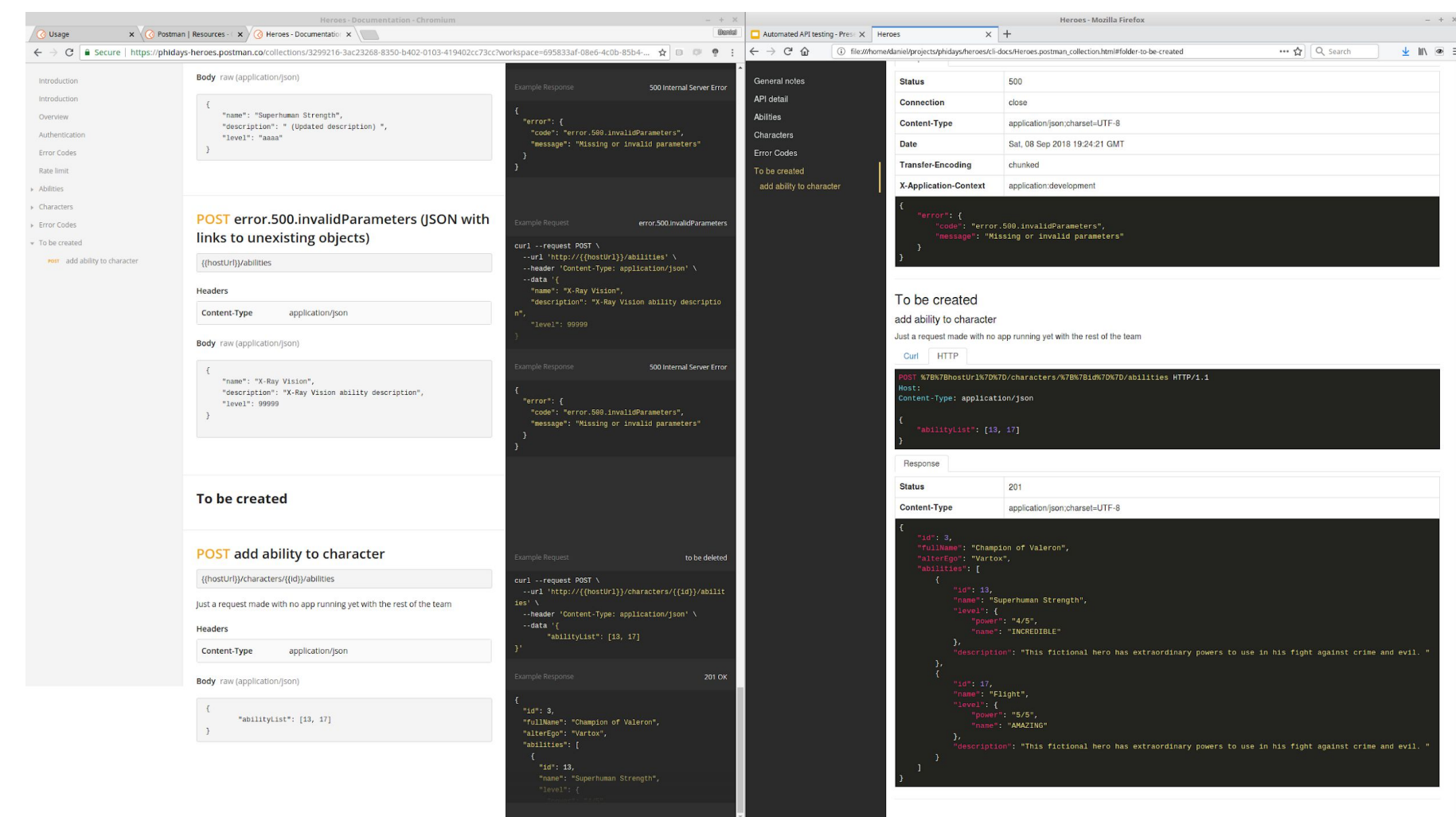
API Documentation from CLI

Run API Tests (Newman) from CLI

Details

Until now, the documentation was
only available at Postman Servers ...

... but now it is possible to be
generated from our servers !!



A hurra for Postmanerator !!





Postman



Direct usage

Share collection.json on gitlab

For back-end & front-end

API Documentation from CLI

Run API Tests (Newman) from CLI

Details

Until now, to run the requests you should use the Postman Client ...

... or run it on the Postman cloud

... but now we can run any collection from our servers !!

A hurra for Newman!!



```
update character
PUT http://localhost:8080//characters/7 [202 Accepted, 732B, 17ms]
✓ response is ok

list characters
GET http://localhost:8080//characters [200 OK, 1.68KB, 10ms]
✓ response is ok

delete character
DELETE http://localhost:8080//characters/7 [204 No Content, 158B, 10ms]
✓ response is ok

Errors
error.404.page.notFound (Bad URL)
DELETE http://localhost:8080//characters_WRONG_URL_/1 [404 Not Found, 256B, 3ms]
✓ Verifying the HTTP status code is correct
✓ Testing the JSON contains the correct content
✓ Response time is less than 200ms

error.404.element.notFound (URL with bad identifier)
PUT http://localhost:8080//characters/9999 [404 Not Found, 276B, 14ms]
✓ response is ok

error.500.missingParameters (JSON missing required param)
POST http://localhost:8080//abilities [500 Internal Server Error, 306B, 49ms]
✓ Verifying the HTTP status code is correct
✓ Testing the JSON contains the correct content
✓ Response time is less than 200ms

error.500.missingParameters (URL missing required param)
PUT http://localhost:8080//abilities/ [500 Internal Server Error, 306B, 8ms]

error.500.invalidParameters (JSON with wrong type param)
PUT http://localhost:8080//abilities/1 [500 Internal Server Error, 306B, 24ms]

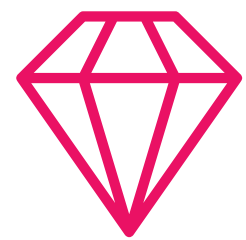
error.500.invalidParameters (JSON with links to unexisting objects)
POST http://localhost:8080//abilities [500 Internal Server Error, 306B, 11ms]
```

	executed	failed
iterations	1	0
requests	13	0
test-scripts	10	0
prerequisite-scripts	0	0
assertions	14	0
total run duration: 543ms		
total data received: 7.26KB (approx)		
average response time: 16ms		

```
daniel@daniel-T460s ~/projects/phidays/heroes/cli-testing$
```



Postman



What if

Begin With the End in Mind

CI/CD API Documentation

CI/CD API Testing

Details

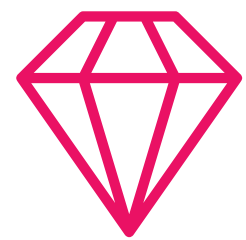
- ❑ Since Postman's API Design Flow makes sense ...
- ❑ What if ...
 - ❑ All parties concerned participates in the **API design from the beginning**, making the first collection of requests a solid base for further developments ?
 - ❑ **Any change is firstly documented** (on branches), by either back or front-end, and then approved and implemented ?

Centralized API Documentation approach!!





Postman



What if

Begin With the End in Mind

CI/CD API Documentation

CI/CD API Testing

Details

- ❑ Since we can ...
 - Generate the API documentation running a command

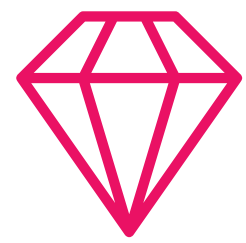
- ❑ What if we integrate it on gitlab CI/CD ?
 - Every pushed commit would generate an HTML with the API doc
 - That could be instantly published in our servers !!

For a public and updated API documentation !!





Postman



What if

Begin With the End in Mind

CI/CD API Documentation

CI/CD API Testing

Details

- ❑ Since we can ...
 - Run the whole API test collection running a command

- ❑ What if we integrate it on gitlab CI/CD ?
 - Every pushed commit would run the complete collection of API tests
 - An email can be sent with the results

No more broken endpoints !!



Try it Yourself!

<https://github.com/daniel-herrero/heroes>

