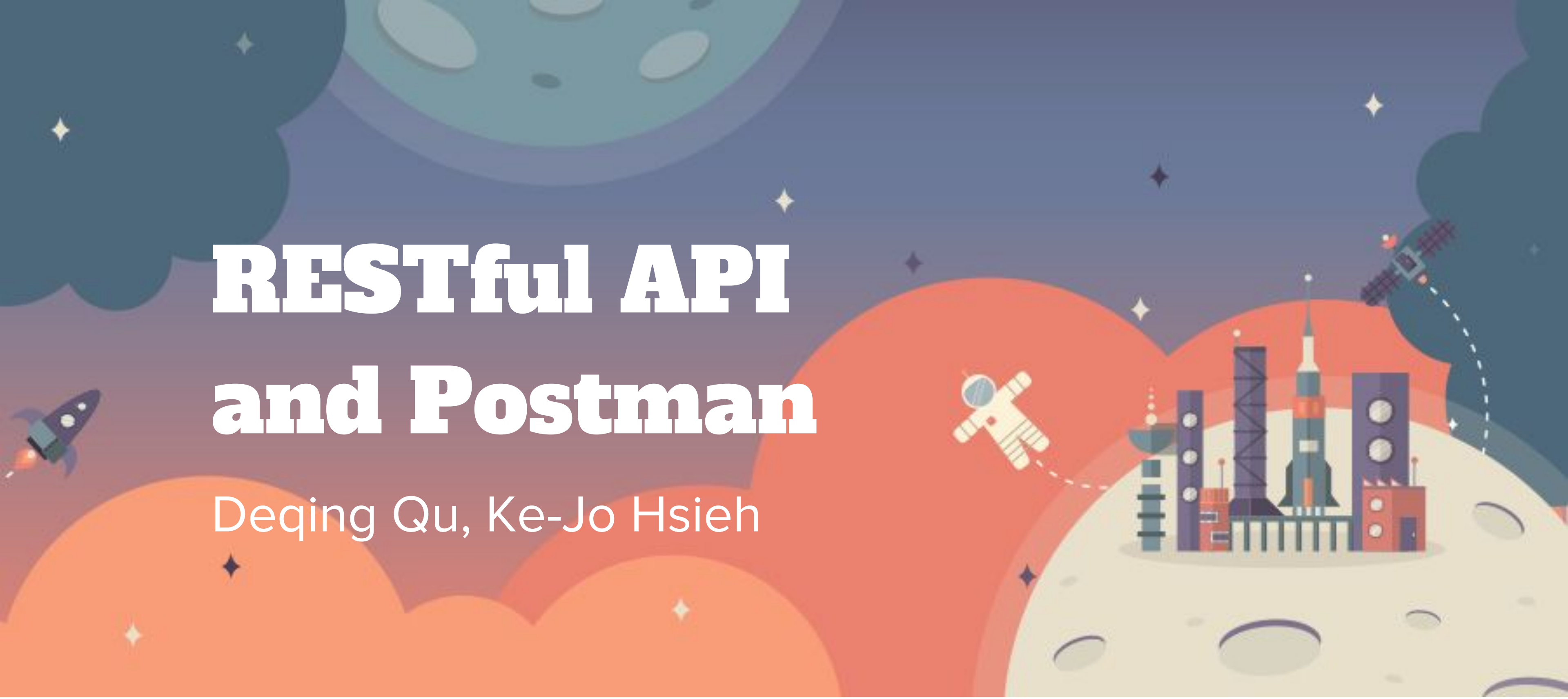


RESTful API and Postman

Deqing Qu, Ke-Jo Hsieh

POSTMAN

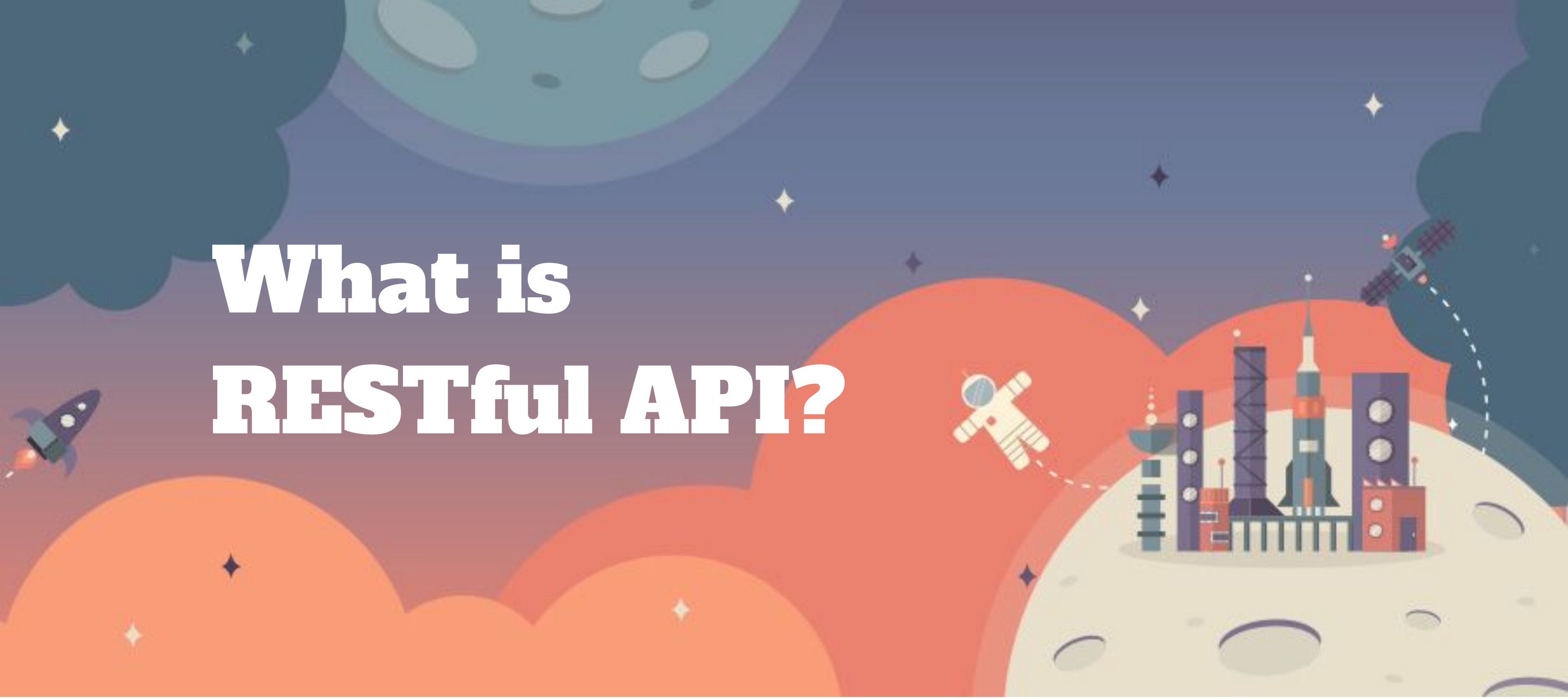
{ REST }

RESTful API

- Intro to RESTful API (20 mins)
- Exercise 0: Design RESTful API (5 mins)

API Test tool - Postman

- Intro to Postman
- Sending API requests (10 mins)
- Exercise 1: First API request (10 mins)
- Scripts (10 mins)
- Exercise 2: test API with scripts (10 mins)
- Environments and globals (10 mins)
- Exercise 3: test with environments (10 mins)
- Collections (10 mins)
- Exercise 4: Make your test cases (20 mins)



What is RESTful API?

{ REST }

Warm-up Questions

{ REST }

1. Have you ever used a third party RESTful API?
2. Have your ever designed a RESTful API?
3. http://example.com/classes/12 or http://example.com/class/12
4. Is it a good idea to store sessions on the server side?

What's RESTful API?

Definition

- REST - Representational State Transfer
- An architecture style by Roy Fielding in 2000 in his dissertation 'Architecture Styles and the Designs of Network Based Architecture'
- a RESTful API - an API follows the REST rules

Core Concept

- **Resources** - All URLs are identified as resources
- **Statelessness** - The RESTful APIs are stateless

Resources

{ REST }

- Every single URL in a RESTful API represents a resource
- **No verbs in URL, only nouns**
- **Use plural in most cases**
- Good Examples
  - GET /classes
  - GET /classes/6
  - GET /classes/6/students
  - GET /classes/6/students/3
- Bad Examples
  - GET /class/6/student/3
  - GET /addStudent

Verbs (Http Request Methods) { REST }

Examples

- GET /classes - Retrieves a list of classes
- GET /classes/12 - Retrieves a specific class
- POST /classes - Create a new class
- PUT /classes/12 - Update class #12
- PATCH /classes/12 - Partially update class #12
- DELETE /classes/12 - Delete classes #12

Designing Guidelines (request) {REST}

How to deal with relations?

- GET /classes/12/students - Retrieves a list of students in class #12
- GET /classes/12/students/2 - Retrieves a specific student in class #12
- POST /classes/12/students/2 - Assign student #2 to class #12
- DELETE /classes/12/students/2 - Remove student #2 from classes #12

What about actions don't fit into CRUD operations?

- PUT /gists/{{id}}/star - star a specific gist
- DELETE /gists/{{id}}/star - unstar a specific gist

Designing Guidelines (request) {REST}

Result sorting and filtering

- GET /classes/12/students?sort=age Retrieves a list of students in class #12 in order of age
- GET /classes/12/students?type=grad Retrieves a list of grad students in class #12

snake_case vs camelCase for field name

- camelCase!

Versioning

- https://example.com/v1/classes/12/students

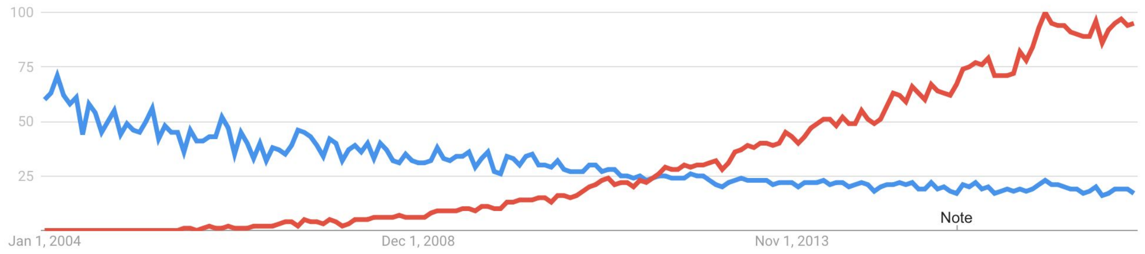
Designing Guidelines (response) { REST }

JSON only response

Search Frequency for:

XML API

JSON API



Response Status Codes

200 OK	201 Created	204 No Content	
304 Not Modified			
400 Bad Request	401 Unauthorized	403 Forbidden	404 Not Found
500 Internal Error			

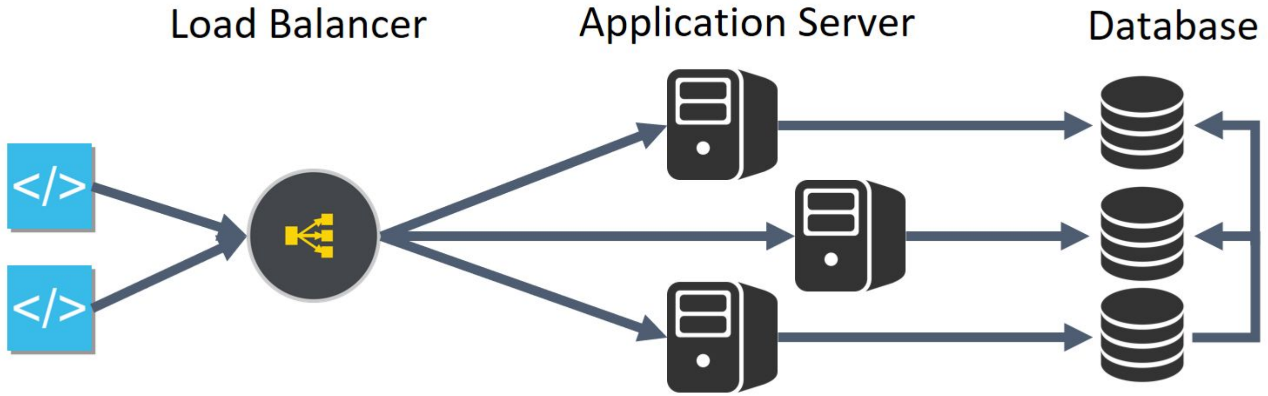
Statelessness

{REST}

Guidelines

- No client data on the server side (don’t use session)
- Any state is maintained on the client side

Why do this? (Scalability)



Exercise 0 - RESTful API design

Assuming you are the API designer of Github, could you design some APIs for the following requirements?

1. List all public repos
2. List all commits on a repo
3. Create a commit comment

Exercise 0 - RESTful API design

1. List all public repos

   GET /repos

2. List all commits on a repo

   GET /repos/{{owner_id}}/{{repo_id}}/commits

3. Create a commit comment

   POST /repos/{{owner_id}}/{{repo_id}}/commits/{{commits_id}}/comments

API Test Tool: Postman

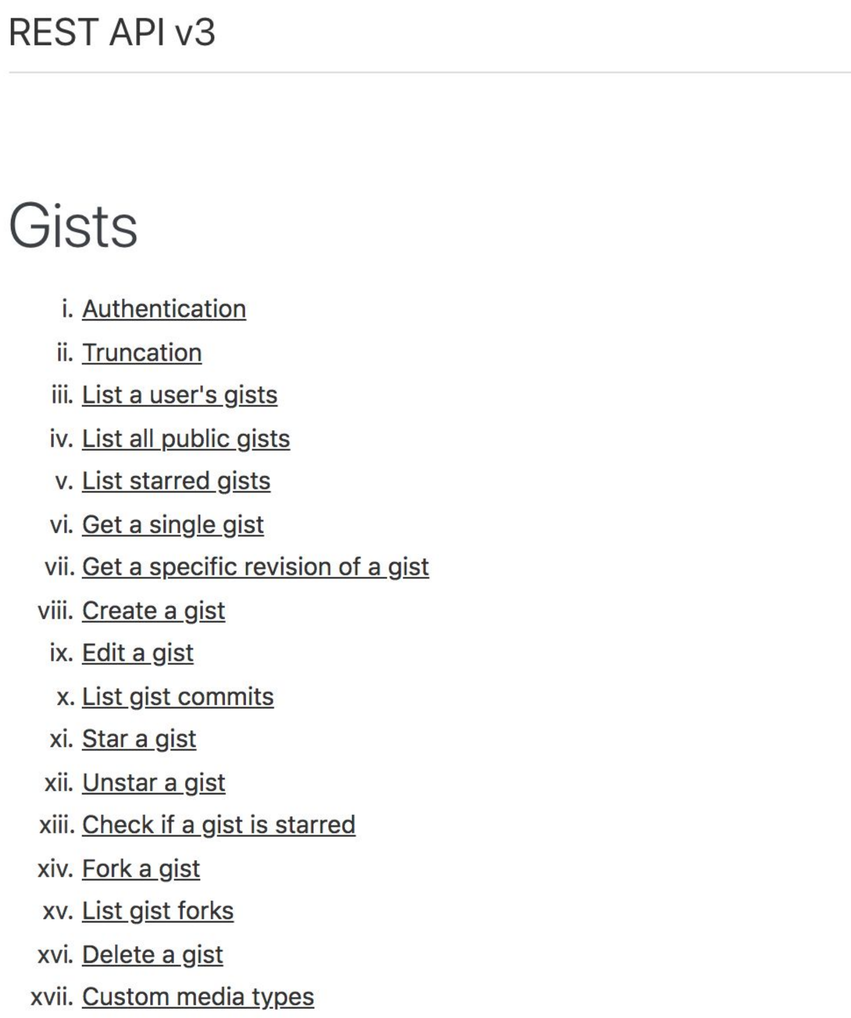
POSTMAN

API Test Tool - Postman

- Automation API Test Tools https://www.getpostman.com/
- Supports OAuth, Cookie, Session ...
- Uses JavaScript to write tests
- Postman is used by 5 million developers and more than 100,000 companies

Github Gists APIs

https://developer.github.com/v3/gists/

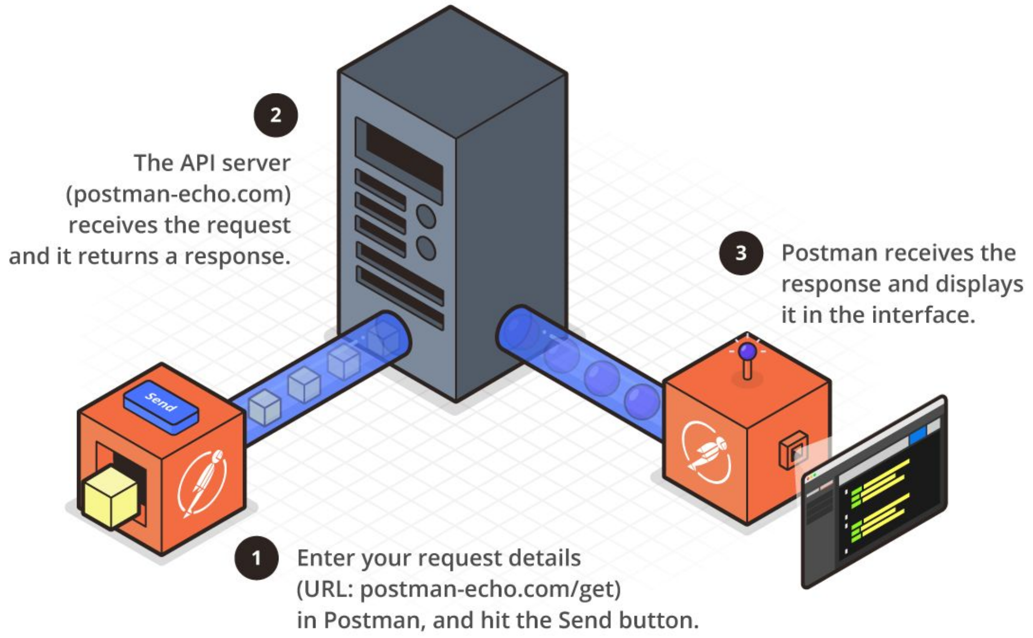


REST API v3

Gists

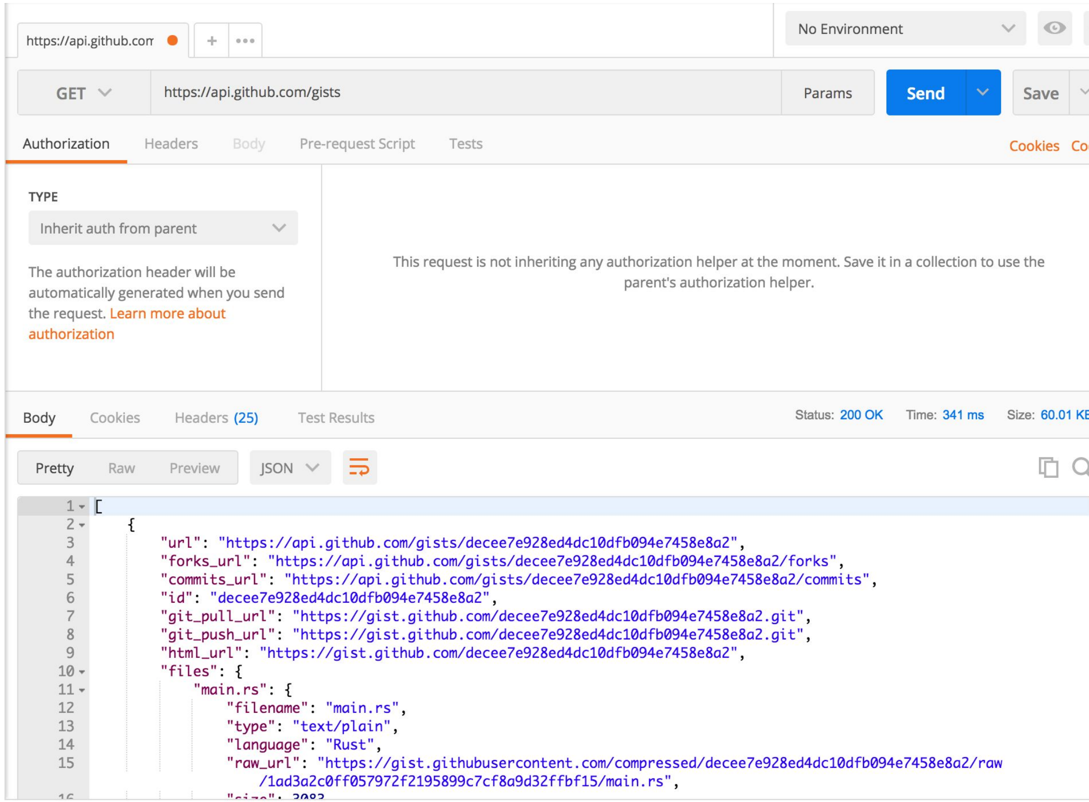
i. Authentication
ii. Truncation
iii. List a user's gists
iv. List all public gists
v. List starred gists
vi. Get a single gist
vii. Get a specific revision of a gist
viii. Create a gist
ix. Edit a gist
x. List gist commits
xi. Star a gist
xii. Unstar a gist
xiii. Check if a gist is starred
xiv. Fork a gist
xv. List gist forks
xvi. Delete a gist
xvii. Custom media types

Sending the first request



Sending the first request

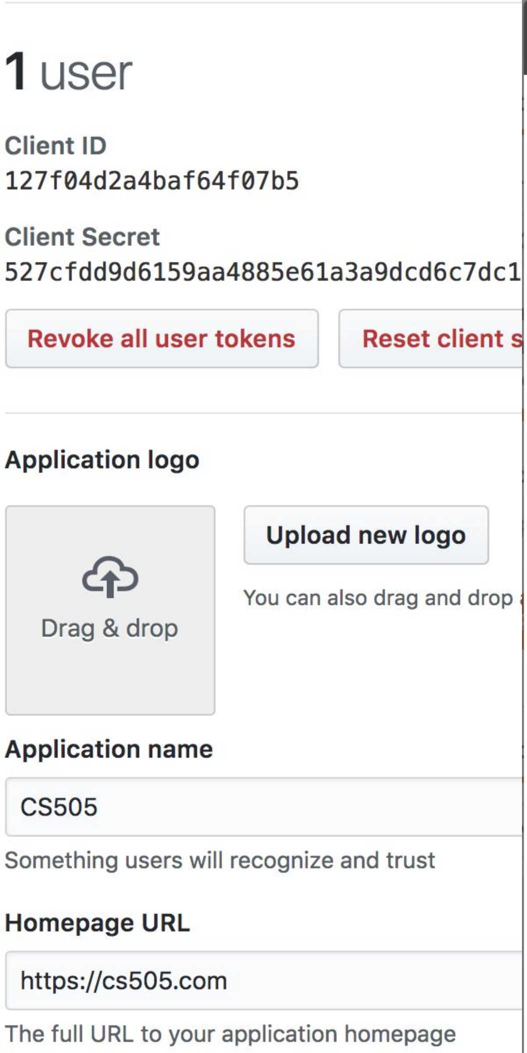
- URL: https://api.github.com/gists
- Method: GET



The authorization header will be automatically generated when you send the request. Learn more about authorization

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

```
[
    {
        "url": "https://api.github.com/gists/decee7e928ed4dc10dfb094e7458e8a2",
        "forks_url": "https://api.github.com/gists/decee7e928ed4dc10dfb094e7458e8a2/forks",
        "commits_url": "https://api.github.com/gists/decee7e928ed4dc10dfb094e7458e8a2/commits",
        "id": "decee7e928ed4dc10dfb094e7458e8a2",
        "git_pull_url": "https://gist.github.com/decee7e928ed4dc10dfb094e7458e8a2.git",
        "git_push_url": "https://gist.github.com/decee7e928ed4dc10dfb094e7458e8a2.git",
        "html_url": "https://gist.github.com/decee7e928ed4dc10dfb094e7458e8a2",
        "files": {
            "main.rs": {
                "filename": "main.rs",
                "type": "text/plain",
                "language": "Rust",
                "raw_url": "https://gist.githubusercontent.com/compressed/decee7e928ed4dc10dfb094e7458e8a2/raw
                    /1ad3a2c0ff057972f2195899c7cf8a9d32ffbf15/main.rs",
                "size": 3083
```


1 user

Client ID
127f04d2a4baf64f07b5

Client Secret
527cfdd9d6159aa4885e61a3a9dcd6c7dc1

Application logo

You can also drag and drop

Application name

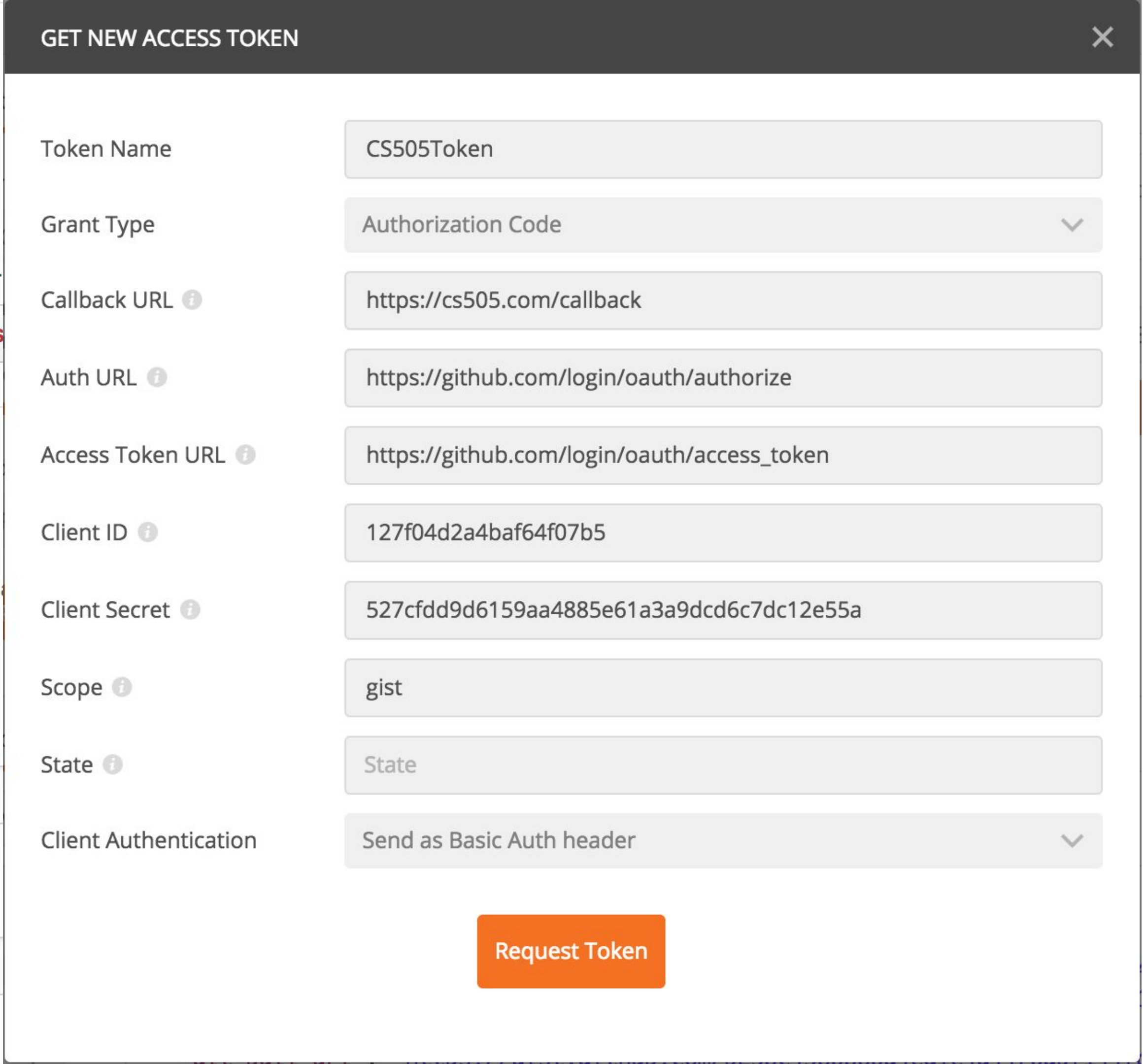
CS505

Something users will recognize and trust

Homepage URL

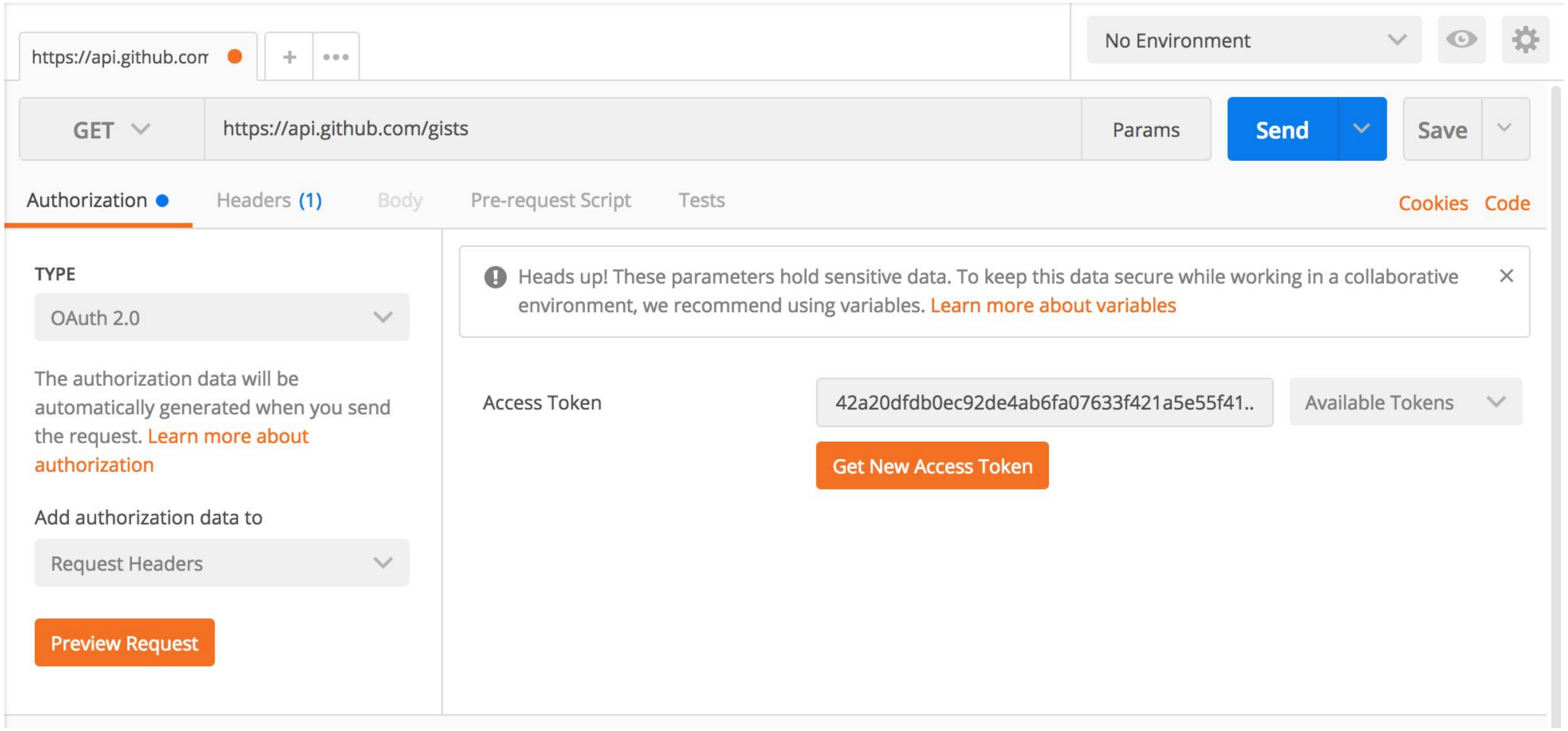
https://cs505.com

The full URL to your application homepage


GET NEW ACCESS TOKEN

Token Name	CS505Token
Grant Type	Authorization Code
Callback URL	https://cs505.com/callback
Auth URL	https://github.com/login/oauth/authorize
Access Token URL	https://github.com/login/oauth/access_token
Client ID	127f04d2a4baf64f07b5
Client Secret	527cfdd9d6159aa4885e61a3a9dcd6c7dc12e55a
Scope	gist
State	State
Client Authentication	Send as Basic Auth header

Authentication


https://api.github.com

No Environment

GET https://api.github.com/gists

TYPE

OAuth 2.0

The authorization data will be automatically generated when you send the request. Learn more about authorization

Add authorization data to

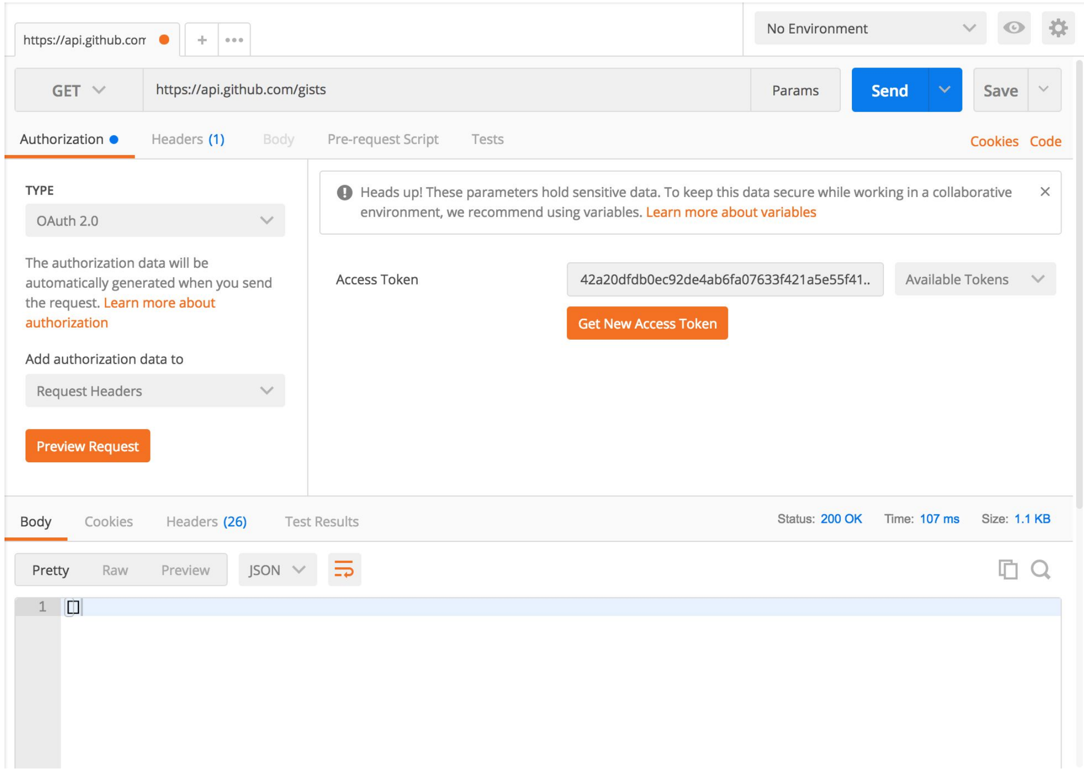
Request Headers

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables

Access Token: 42a20dfdb0ec92de4ab6fa07633f421a5e55f41..

Sending the same GET request

- URL: https://api.github.com/gists
- Method: GET
- Authorization: OAuth2



The authorization data will be automatically generated when you send the request. Learn more about authorization

Add authorization data to: Request Headers

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables

Access Token: 42a20dfdb0ec92de4ab6fa07633f421a5e55f41..

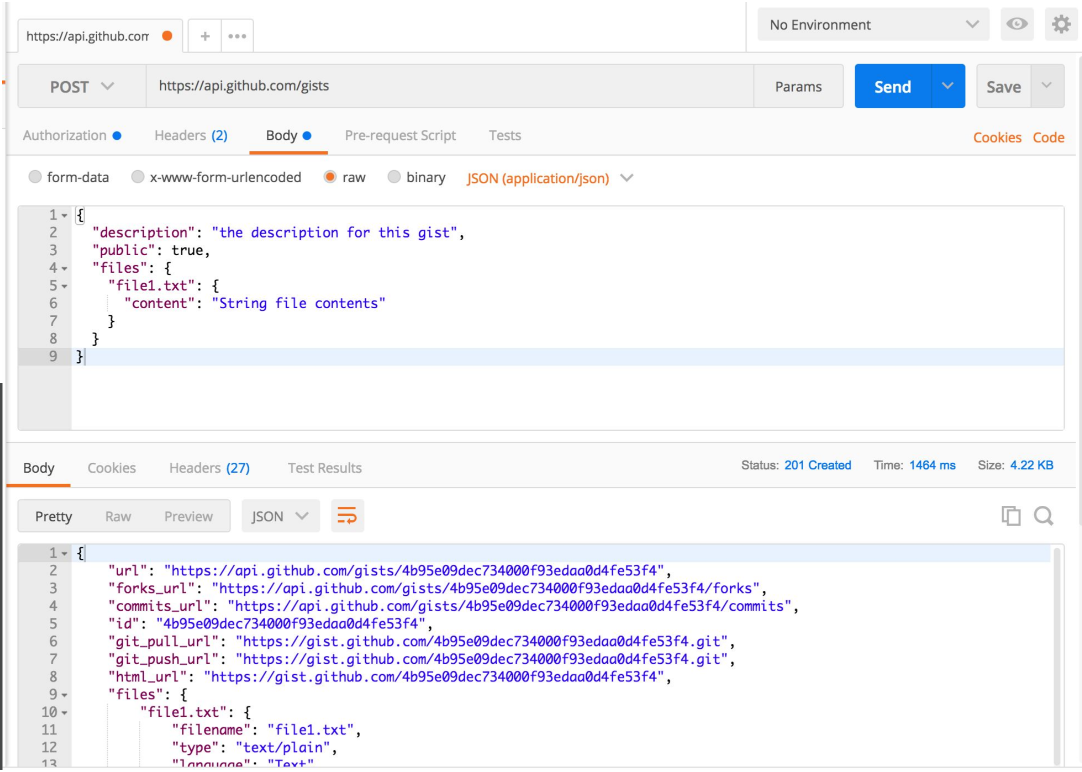
Status: 200 OK Time: 107 ms Size: 1.1 KB

```
[]
```

Sending a POST request

- URL: https://api.github.com/gists
- Method: POST
- Authorization: OAuth2
- Body (JSON format)

```
{
  "description": "the description for this gist",
  "public": true,
  "files": {
    "file1.txt": {
      "content": "String file contents"
    }
  }
}
```



https://api.github.com | No Environment

POST https://api.github.com/gists | Params | Send | Save

Authorization | Headers (2) | Body | Pre-request Script | Tests | Cookies | Code

form-data | x-www-form-urlencoded | [✓ raw] | binary | JSON (application/json)

```
{
  "description": "the description for this gist",
  "public": true,
  "files": {
    "file1.txt": {
      "content": "String file contents"
    }
  }
}
```

Body | Cookies | Headers (27) | Test Results — Status: 201 Created Time: 1464 ms Size: 4.22 KB

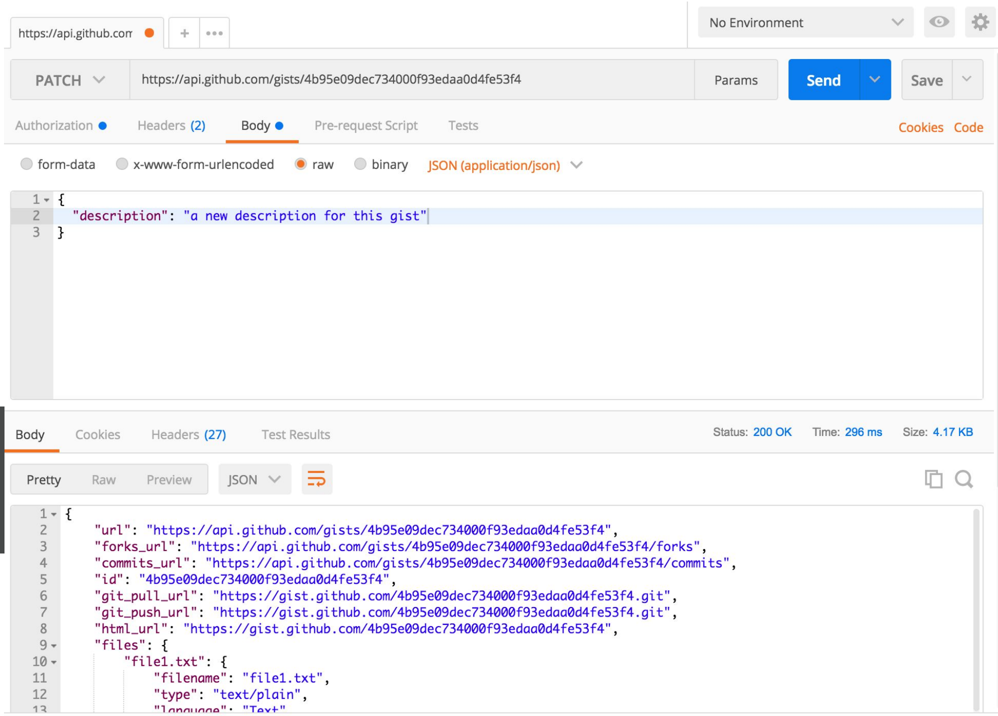
Pretty | Raw | Preview | JSON

```
{
    "url": "https://api.github.com/gists/4b95e09dec734000f93edaa0d4fe53f4",
    "forks_url": "https://api.github.com/gists/4b95e09dec734000f93edaa0d4fe53f4/forks",
    "commits_url": "https://api.github.com/gists/4b95e09dec734000f93edaa0d4fe53f4/commits",
    "id": "4b95e09dec734000f93edaa0d4fe53f4",
    "git_pull_url": "https://gist.github.com/4b95e09dec734000f93edaa0d4fe53f4.git",
    "git_push_url": "https://gist.github.com/4b95e09dec734000f93edaa0d4fe53f4.git",
    "html_url": "https://gist.github.com/4b95e09dec734000f93edaa0d4fe53f4",
    "files": {
        "file1.txt": {
            "filename": "file1.txt",
            "type": "text/plain",
            "language": "Text"
```

Sending a PATCH request

- URL: https://api.github.com/gists/{{id}}
- Method: PATCH
- Authorization: OAuth2
- Body (JSON format)

```
{
  "description": "a new description for this gist"
}
```



PATCH https://api.github.com/gists/4b95e09dec734000f93edaa0d4fe53f4

Body: raw, JSON (application/json)

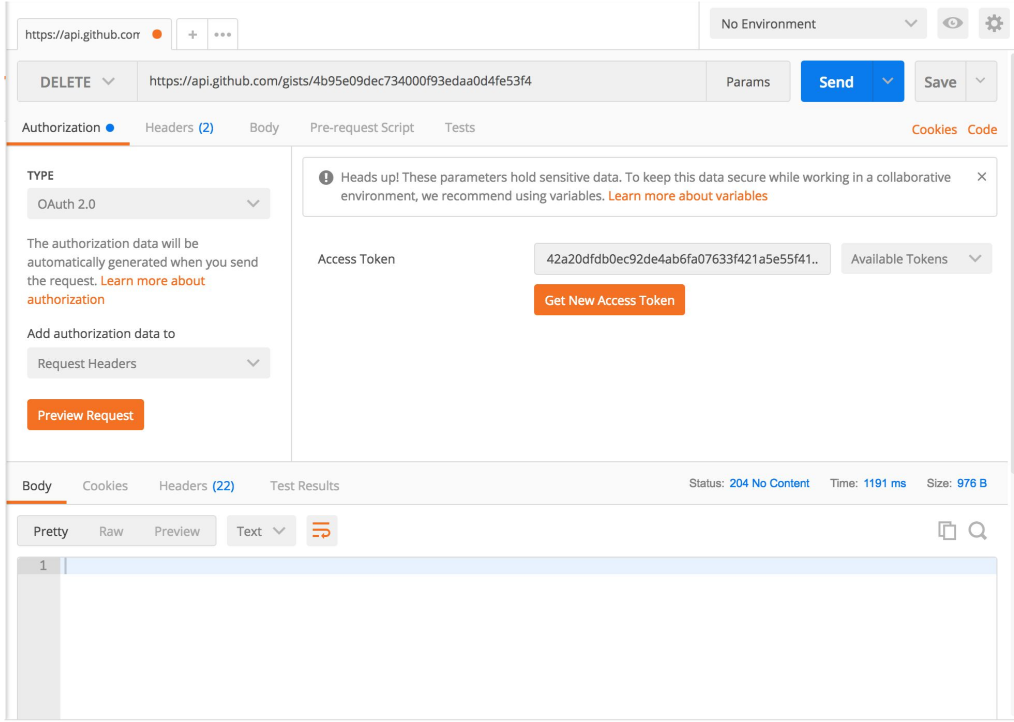
```
{
  "description": "a new description for this gist"
}
```

Status: 200 OK Time: 296 ms Size: 4.17 KB

```
{
    "url": "https://api.github.com/gists/4b95e09dec734000f93edaa0d4fe53f4",
    "forks_url": "https://api.github.com/gists/4b95e09dec734000f93edaa0d4fe53f4/forks",
    "commits_url": "https://api.github.com/gists/4b95e09dec734000f93edaa0d4fe53f4/commits",
    "id": "4b95e09dec734000f93edaa0d4fe53f4",
    "git_pull_url": "https://gist.github.com/4b95e09dec734000f93edaa0d4fe53f4.git",
    "git_push_url": "https://gist.github.com/4b95e09dec734000f93edaa0d4fe53f4.git",
    "html_url": "https://gist.github.com/4b95e09dec734000f93edaa0d4fe53f4",
    "files": {
        "file1.txt": {
            "filename": "file1.txt",
            "type": "text/plain",
            "language": "Text"
```

Sending a DELETE request

- URL: https://api.github.com/gists/{{id}}
- Method: DELETE
- Authorization: OAuth2



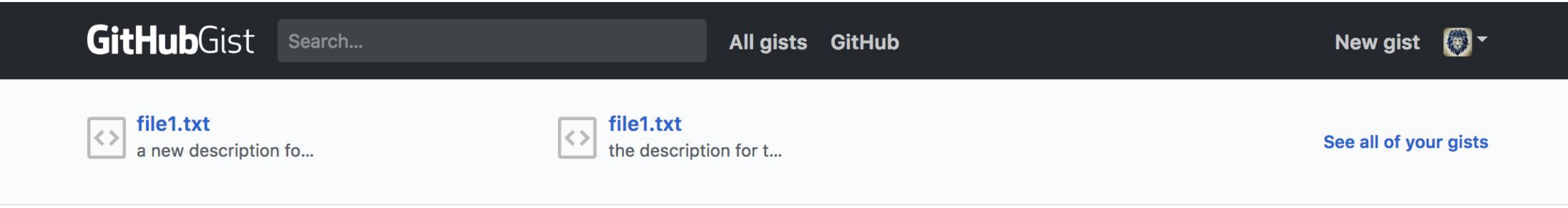
The authorization data will be automatically generated when you send the request. Learn more about authorization

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables

Status: 204 No Content Time: 1191 ms Size: 976 B

Exercise 1 - sending requests

- Create 4 requests:
  - Get all Gists
  - Create a new Gist
  - Modify the Gist
  - Delete the Gist
- Check the results on https://gist.github.com/



file1.txt
a new description fo...

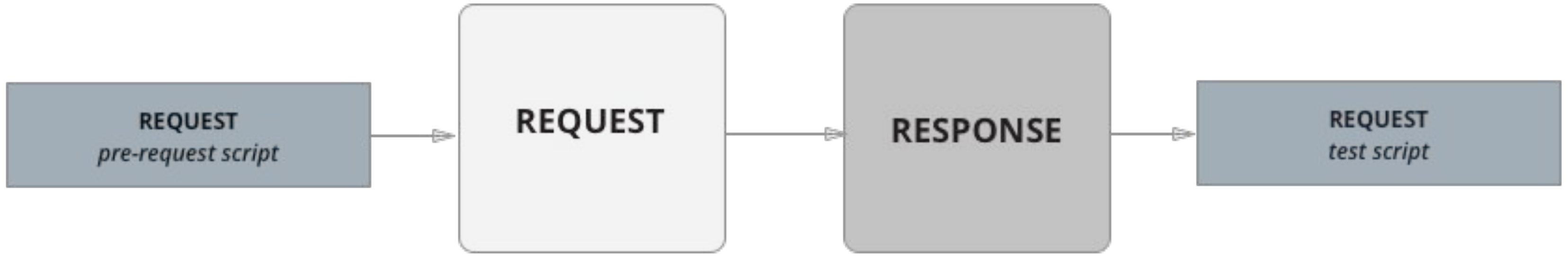
file1.txt
the description for t...

Postman Test Scripts

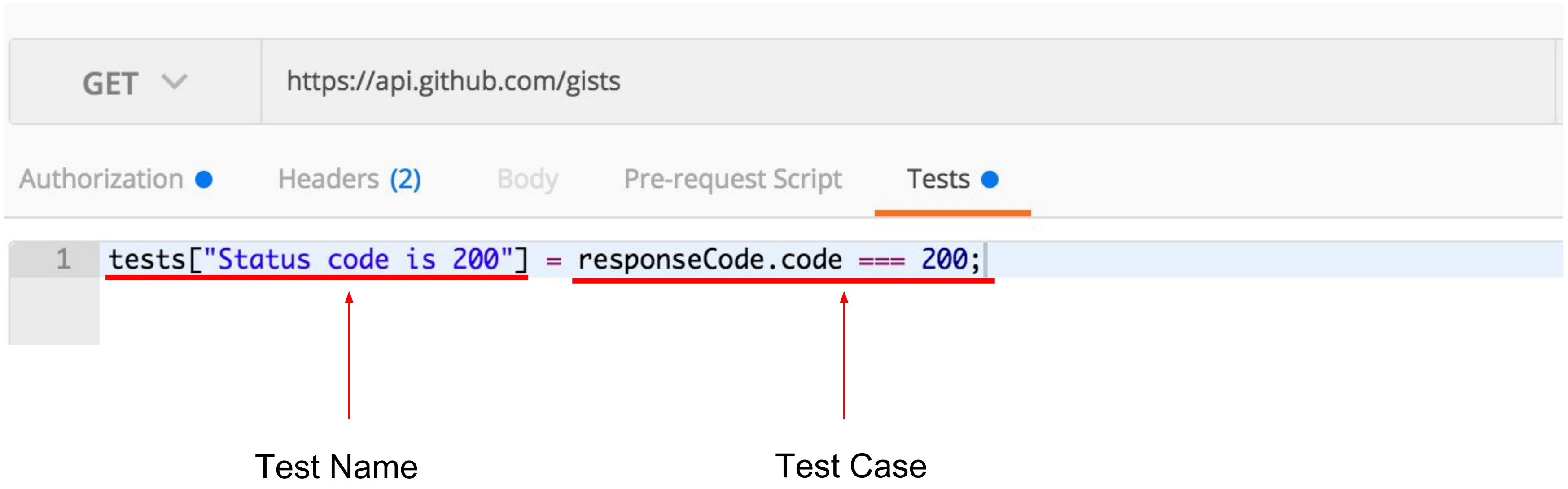
POSTMAN

Postman Automated Test Scripts

- Automatically checks your server's response
- Javascript
- Could be executed:
  - **Before** the request in send to the server (**Pre-request Script**)
  - **After** response has been received (**Test**)

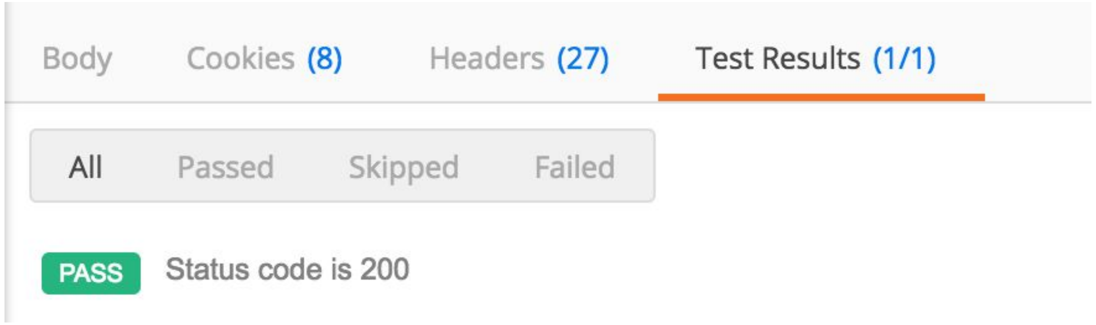


Example (Testing Status Code)



```
tests["Status code is 200"] = responseCode.code === 200;
```

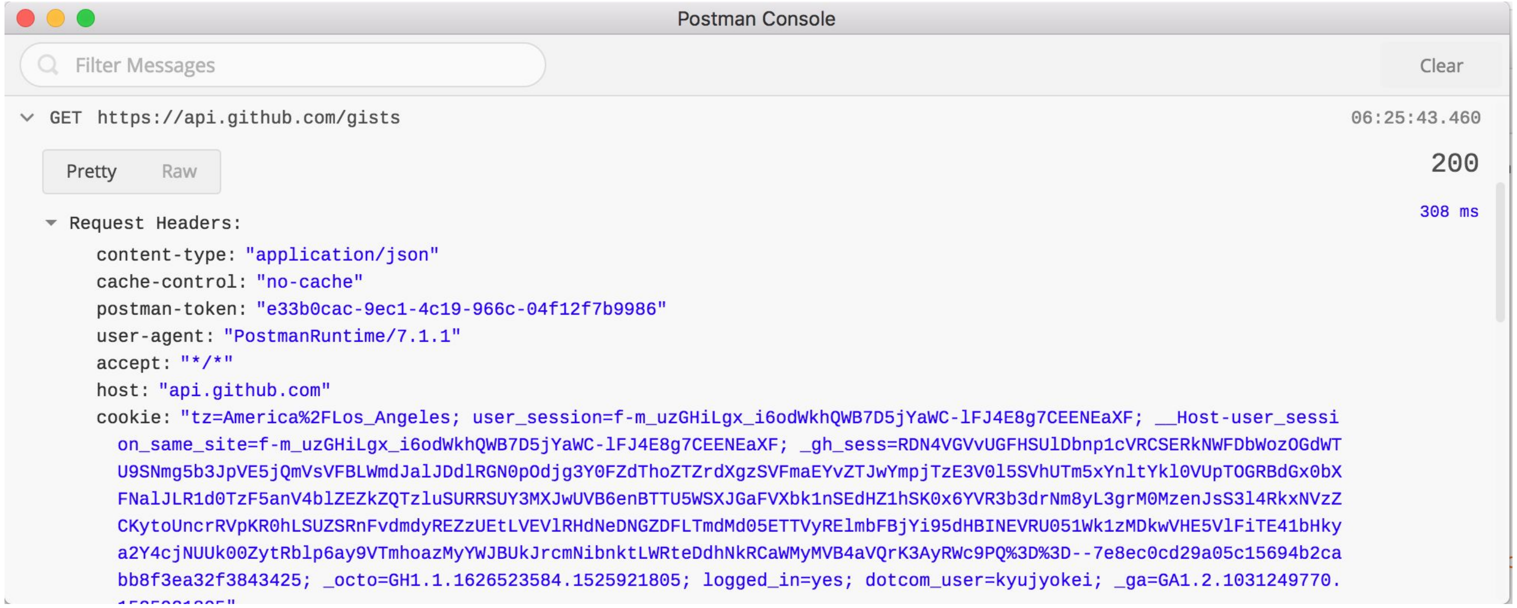
Status Code



Status code is 200

Console.log()

Show Postman Console: (CMD/CTRL + ALT + C)

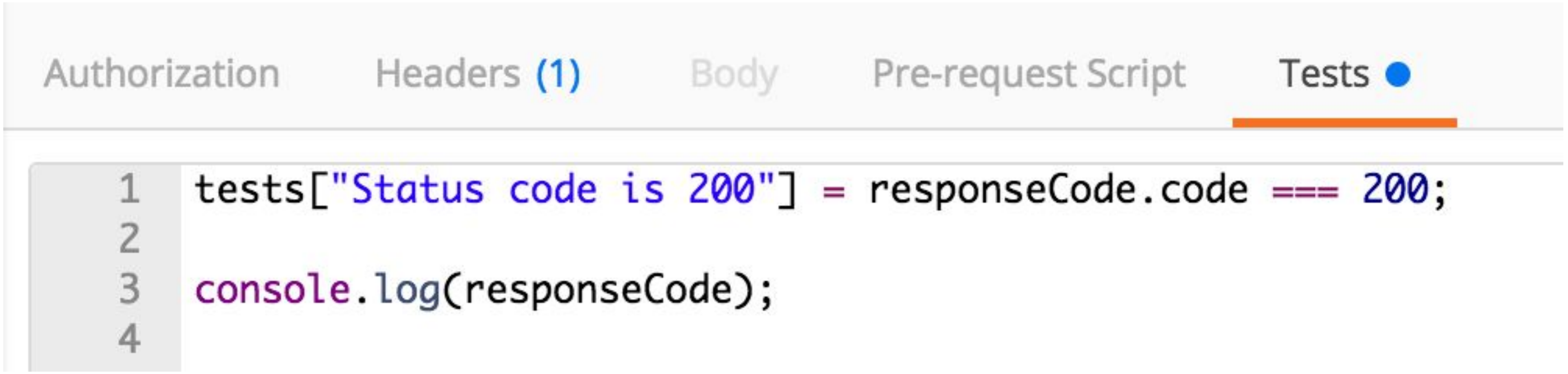


```
GET  https://api.github.com/gists                                                   06:25:43.460
                                                                                              200
                                                                                           308 ms
  Request Headers:
    content-type: "application/json"
    cache-control: "no-cache"
    postman-token: "e33b0cac-9ec1-4c19-966c-04f12f7b9986"
    user-agent: "PostmanRuntime/7.1.1"
    accept: "*/*"
    host: "api.github.com"
    cookie: "tz=America%2FLos_Angeles; user_session=f-m_uzGHiLgx_i6odWkhQWB7D5jYaWC-lFJ4E8g7CEENEaXF; __Host-user_sessi
      on_same_site=f-m_uzGHiLgx_i6odWkhQWB7D5jYaWC-lFJ4E8g7CEENEaXF; _gh_sess=RDN4VGVvUGFHSUlDbnp1cVRCSERkNWFDbWozOGdWT
      U9SNmg5b3JpVE5jQmVsVFBLWmdJalJDdlRGN0pOdjg3Y0FZdThoZTZrdXgzSVFmaEYvZTJwYmpjTzE3V0l5SVhUTm5xYnltYkl0VUpTOGRBdGx0bX
      FNalJLR1d0TzF5anV4blZEZkZQTzluSURRSUY3MXJwUVB6enBTTU5WSXJGaFVXbk1nSEdHZ1hSK0x6YVR3b3drNm8yL3grM0MzenJsS3l4RkxNVzZ
      CKytoUncrRVpKR0hLSUZSRnFvdmdyREZzUEtLVEVlRHdNeDNGZDFLTmdMd05ETTVyRElmbFBjYi95dHBINEVRU051Wk1zMDkwVHE5VlFiTE41bHky
      a2Y4cjNUUk00ZytRblp6ay9VTmhoazMyYWJBUkJrcmNibnktLWRteDdhNkRCaWMyMVB4aVQrK3AyRWc9PQ%3D%3D--7e8ec0cd29a05c15694b2ca
      bb8f3ea32f3843425; _octo=GH1.1.1626523584.1525921805; logged_in=yes; dotcom_user=kyujyokei; _ga=GA1.2.1031249770.
      1525921805"
```

Exercise 2 - testing the requests

Test if the status code is 200, and use Console.log() to print out responseCode

What’s inside response Code?



```
tests["Status code is 200"] = responseCode.code === 200;

console.log(responseCode);

```

Check Content Type

```
var contentTypeHeaderExists = responseHeaders.hasOwnProperty("Content-Type");

tests["Has Content-Type"] = contentTypeHeaderExists;

if (contentTypeHeaderExists) {
    tests["Content-Type is application/json"] =
responseHeaders["Content-Type"].has("application/json");
}
```

Counts of Response

Check if the public Gists returns 30 Gists.

- URL: https://api.github.com/gists
- Method: GET
- Test:

```
responseJson = JSON.parse(responseBody);
tests["Expected number"] = responseJson.length === 30;
```

Search for a Certain Response

- URL: https://api.github.com/gists
- Method: POST
- Authorization: OAuth2
- Body (JSON format)
- Test

```json
{
  "description": "The ice cream flavors that I like.",
  "public": true,
  "files": {
    "file0.txt": {
      "content": "Chocolate, Strawberry, Pacific Cod"
    }
  }
}
```

```javascript
tests["Has Chocolate"] = responseBody.has("Chocolate");
```

Globals - Set Global Variables

Provide a set of variables that are always available to you in all scopes.

Set a Global value:

```
pm.globals.set("variable_key", variable_value);
```

Get your Global value:

```
pm.globals.get("variable_key");
```

Unset Global value:

```
pm.globals.unset("variable_key");
```

Globals - Example

1. Set Global variable in test code while GET

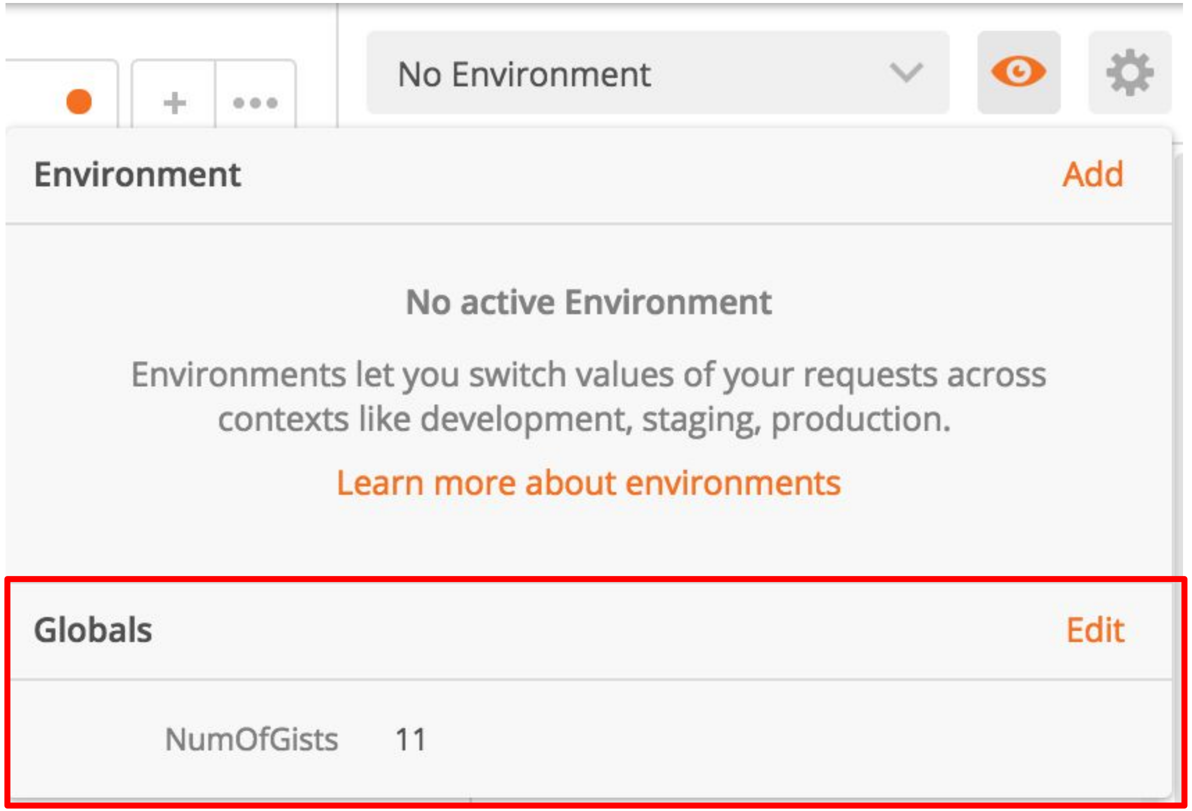
```
responseJson = JSON.parse(responseBody);
pm.globals.set("num_of_gists", responseJson.length);
```

2. Post another Gist

3. Test if total Gists count increased by 1

```
responseJson = JSON.parse(responseBody);
tests["Gists Increased by 1"] = responseJson.length === pm.globals.get("num_of_gists") + 1;
```

Globals - View & Edit



Environment Add

No active Environment

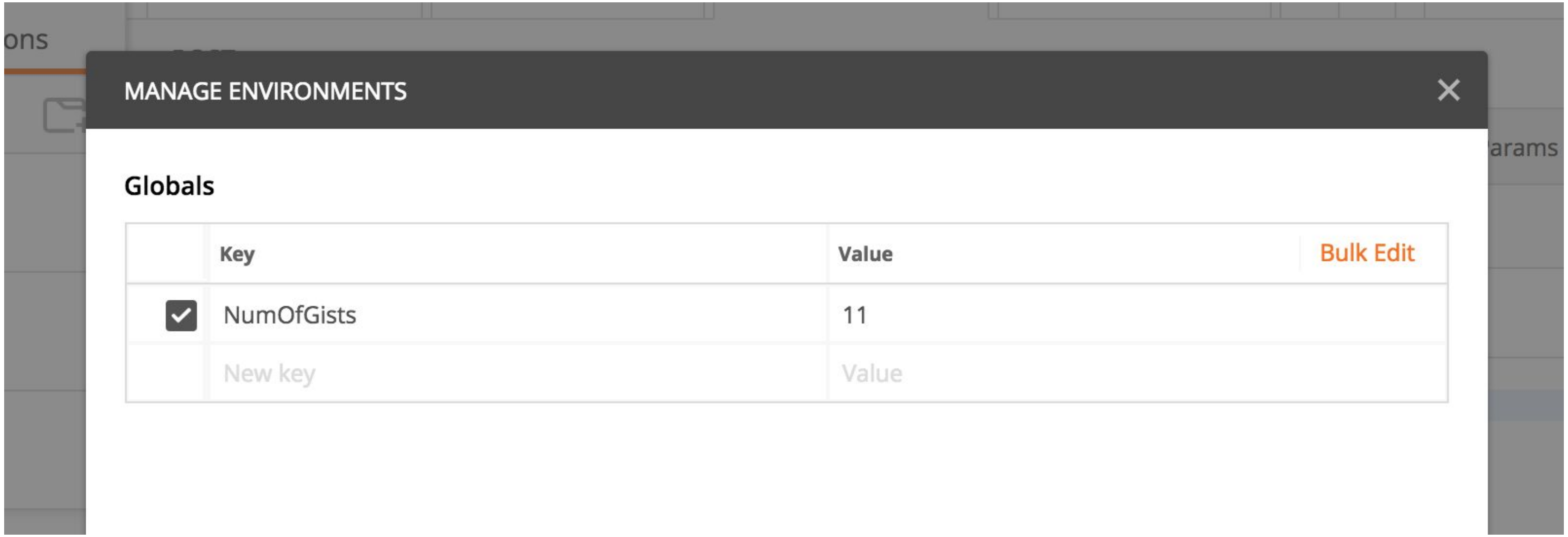
Environments let you switch values of your requests across contexts like development, staging, production.

Learn more about environments

Globals Edit

NumOfGists	11

Globals - View & Edit

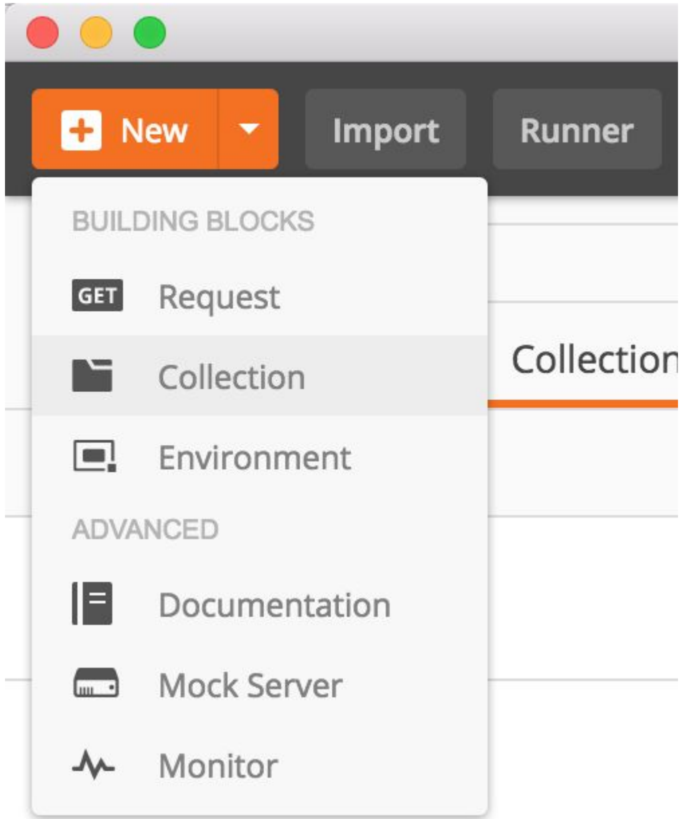


Globals

	Key	Value
✓	NumOfGists	11
	New key	Value

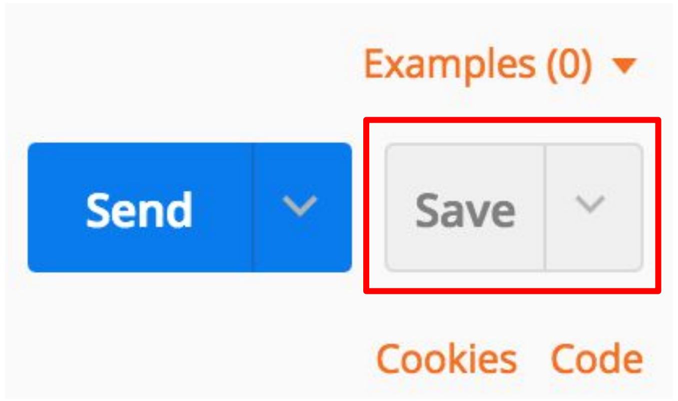
Collections

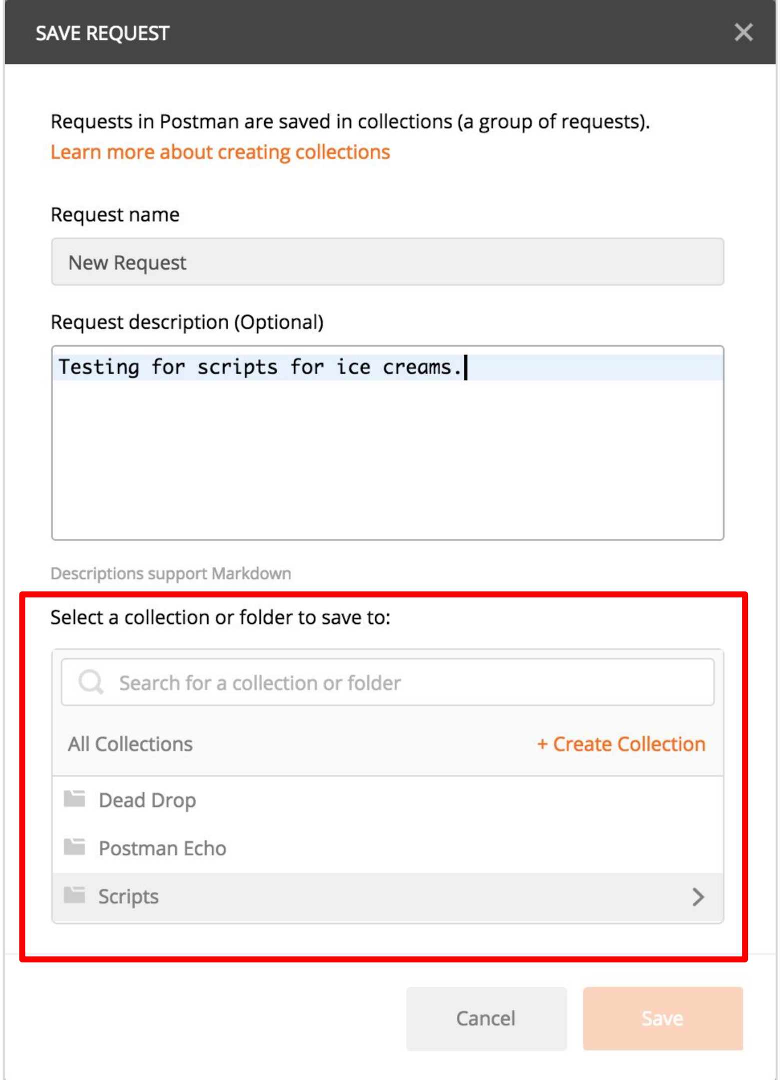
- Collections are groups of requests that can be run together as a series.
- Useful when you want to automate API testing.
- When you use scripts, you can build integration test suites, pass data between API requests, and build workflows that mirror your actual use case of APIs.



Save to Collections

Save your requests in Collections:




SAVE REQUEST

Requests in Postman are saved in collections (a group of requests).
Learn more about creating collections

Request name

New Request

Request description (Optional)

```
Testing for scripts for ice creams.
```

Descriptions support Markdown

Select a collection or folder to save to:

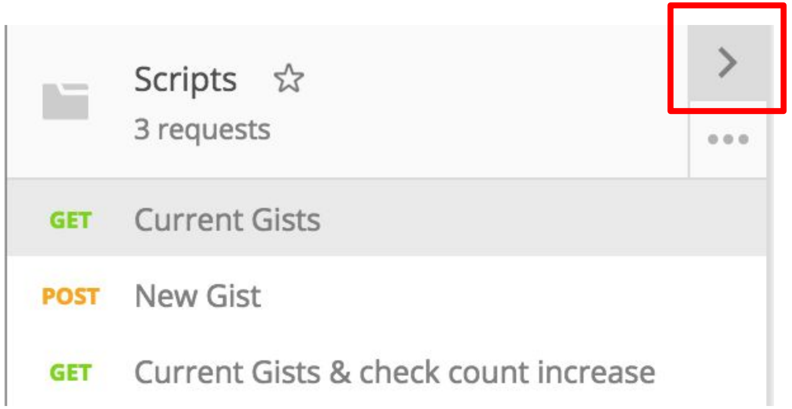
Search for a collection or folder

All Collections + Create Collection

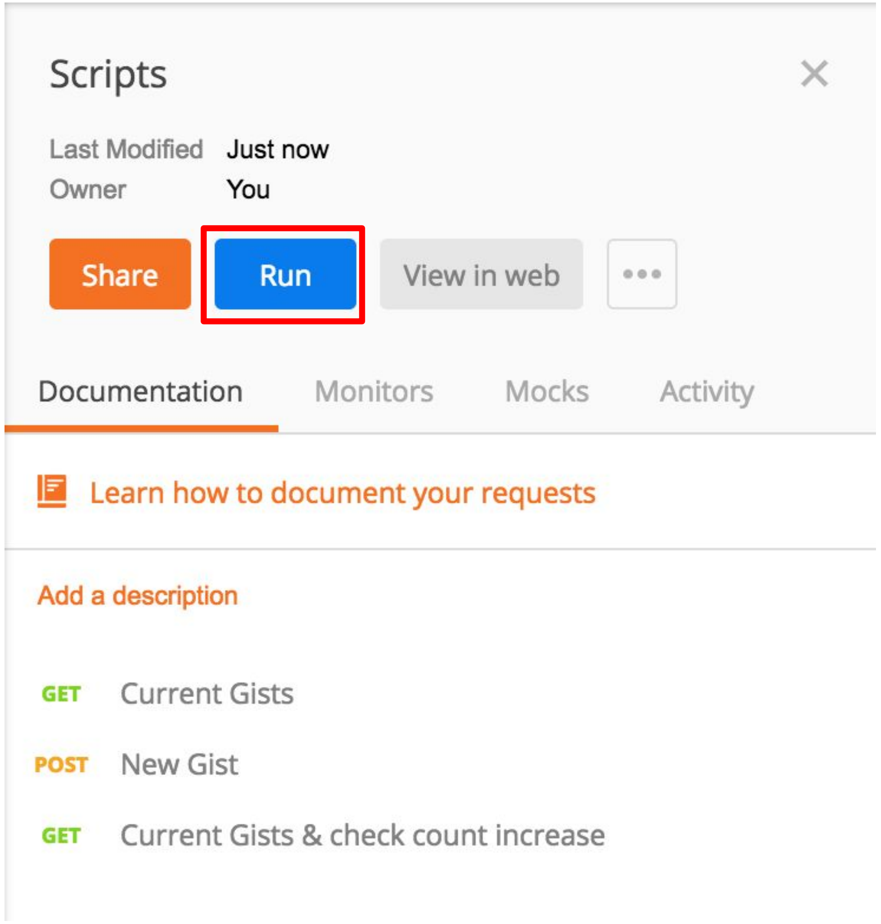
- Dead Drop
- Postman Echo
- Scripts

Run Collection

1. Hit the > button in Collection
2. Click "Run"

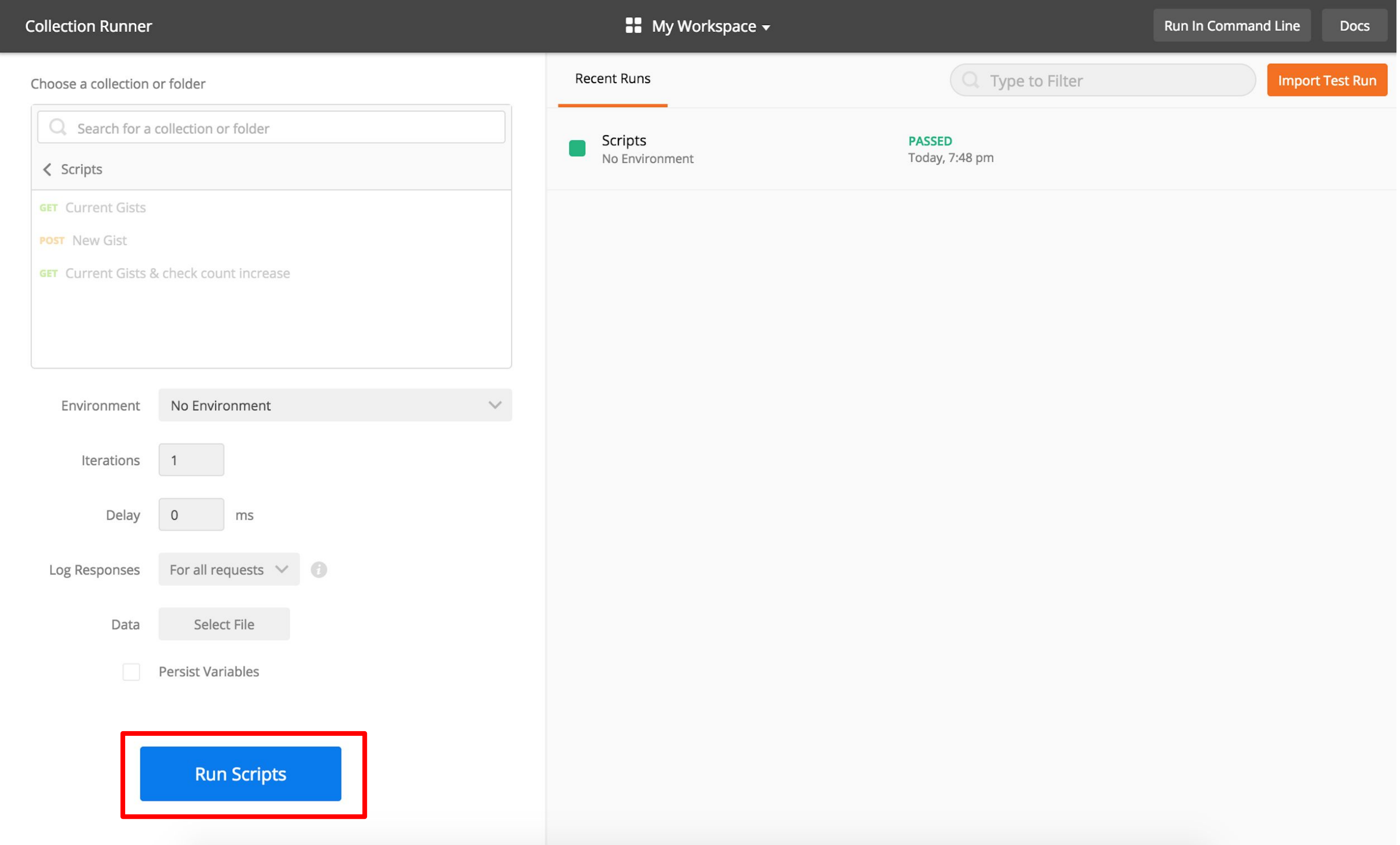


GET Current Gists
POST New Gist
GET Current Gists & check count increase



Last Modified Just now
Owner You

GET Current Gists
POST New Gist
GET Current Gists & check count increase



Collection Runner

My Workspace

Run In Command Line

Docs

Choose a collection or folder

Search for a collection or folder

Scripts

GET Current Gists

POST New Gist

GET Current Gists & check count increase

Environment: No Environment

Iterations: 1

Delay: 0 ms

Log Responses: For all requests

Data: Select File

Persist Variables

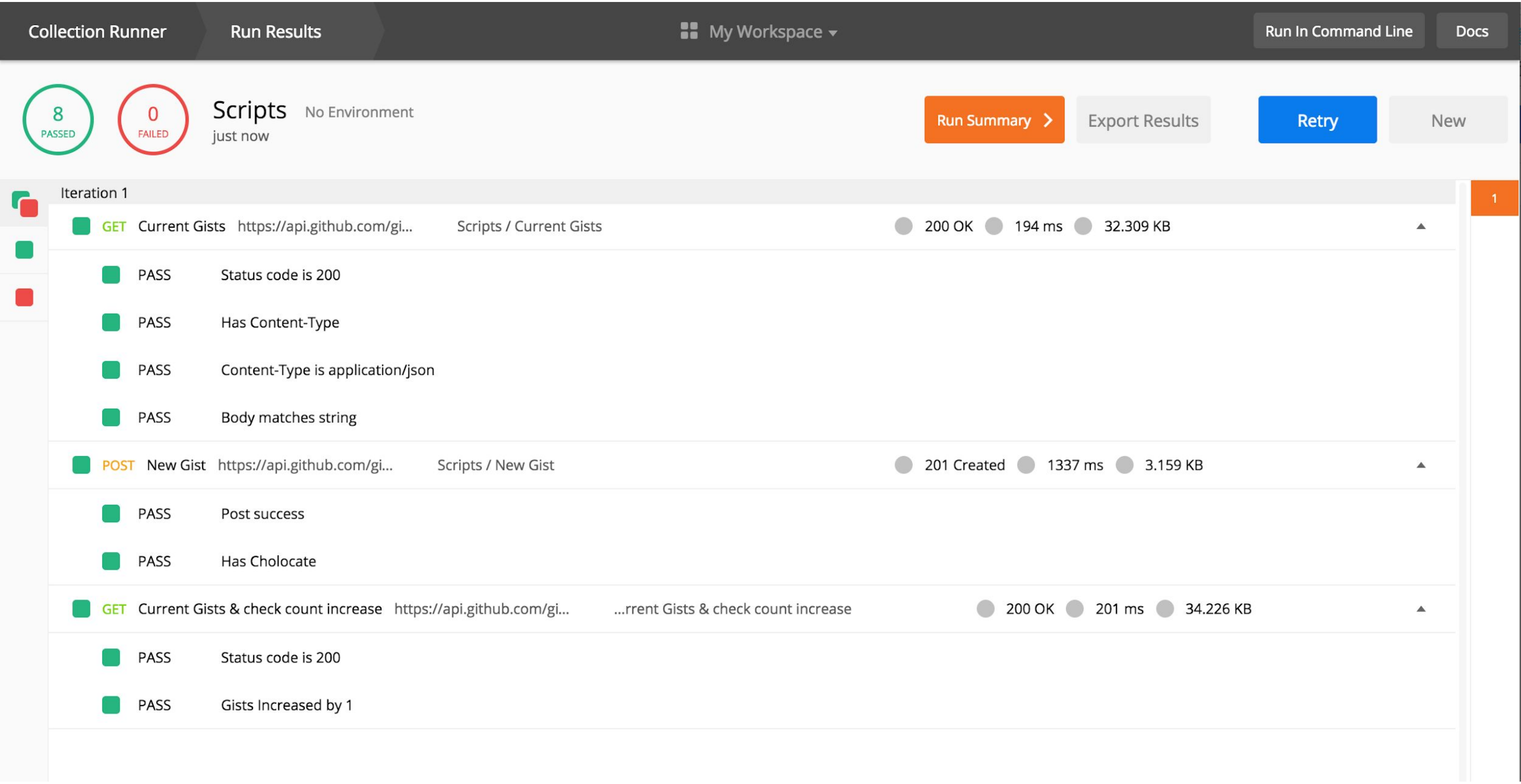
Run Scripts

Recent Runs

Type to Filter

Import Test Run

🟩	Scripts No Environment	PASSED Today, 7:48 pm


Collection Runner | Run Results | My Workspace | Run In Command Line | Docs

8 PASSED | 0 FAILED | Scripts No Environment | just now | Run Summary | Export Results | Retry | New

Iteration 1

GET Current Gists https://api.github.com/gi... Scripts / Current Gists 200 OK 194 ms 32.309 KB

PASS Status code is 200

PASS Has Content-Type

PASS Content-Type is application/json

PASS Body matches string

POST New Gist https://api.github.com/gi... Scripts / New Gist 201 Created 1337 ms 3.159 KB

PASS Post success

PASS Has Cholocate

GET Current Gists & check count increase https://api.github.com/gi... ...rrent Gists & check count increase 200 OK 201 ms 34.226 KB

PASS Status code is 200

PASS Gists Increased by 1

Exercise 4 - building your test suite

1. Confirm that when you create a Gist the number of Gists associated to your account increases by 1
2. Confirm that the contents of the Gist you created match the contents you sent
3. Confirm that you are able to edit the contents of a Gist (this will require editing it and proving the edits worked)
4. Confirm you can delete a Gist

Solution Part 1

1. GET /gists/

```
tests["Status code is 200"] = responseCode.code === 200;

responseJson = JSON.parse(responseBody);
pm.globals.set("num_of_gists", responseJson.length);
```

2. POST /gists/

```
tests["Status code is 201"] = responseCode.code === 201;

tests["Has Content"] = responseBody.has("Your content");

responseJson = JSON.parse(responseBody);
pm.globals.set("gist_id", responseJson.id);
```

Solution Part 2

3. GET /gists/

```
tests["Status code is 200"] = responseCode.code === 200;

responseJson = JSON.parse(responseBody);
tests["Gists count Increased by 1"] = responseJson.length ===
pm.globals.get("num_of_gists") + 1;
```

Solution Part 3

4. PATCH /gists/{{gist_id}}

```
tests["Status code is 200"] = responseCode.code === 200;

tests["Body matches string"] = responseBody.has("Changed Context");
```

5.GET /gists/{{gist_id}}

```
tests["Status code is 200"] = responseCode.code === 200;

responseJson = JSON.parse(responseBody);
tests["JSON Body matches string"] = responseJson['files']['file1.txt']['content']
=== "Changed Context"
```

Solution Part 4

6. DELETE /gists/{{gist_id}}

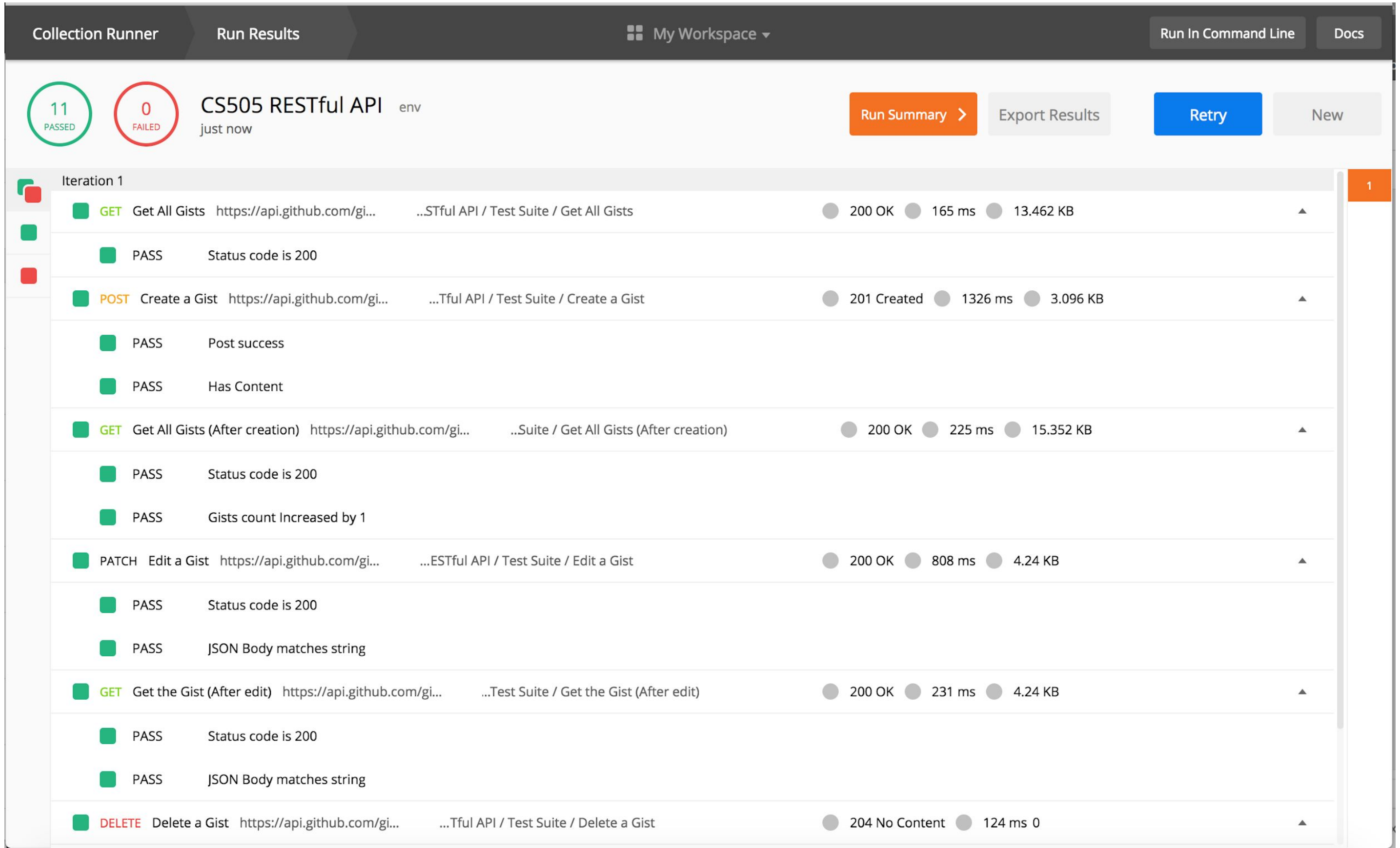
```
tests["Status code is 204"] = responseCode.code === 204;
```

7. GET /gists/{{gist_id}}

```
tests["Status code is 404"] = responseCode.code === 404;

pm.globals.unset("gist_id");
```

All Test Cases Passed!



11 PASSED 0 FAILED

CS505 RESTful API env
just now

Iteration 1

```
GET     Get All Gists  https://api.github.com/gi...  ...STful API / Test Suite / Get All Gists  200 OK  165 ms  13.462 KB
        PASS   Status code is 200
POST    Create a Gist  https://api.github.com/gi...  ...Tful API / Test Suite / Create a Gist  201 Created  1326 ms  3.096 KB
        PASS   Post success
        PASS   Has Content
GET     Get All Gists (After creation)  https://api.github.com/gi...  ..Suite / Get All Gists (After creation)  200 OK  225 ms  15.352 KB
        PASS   Status code is 200
        PASS   Gists count Increased by 1
PATCH   Edit a Gist  https://api.github.com/gi...  ...ESTful API / Test Suite / Edit a Gist  200 OK  808 ms  4.24 KB
        PASS   Status code is 200
        PASS   JSON Body matches string
GET     Get the Gist (After edit)  https://api.github.com/gi...  ...Test Suite / Get the Gist (After edit)  200 OK  231 ms  4.24 KB
        PASS   Status code is 200
        PASS   JSON Body matches string
DELETE  Delete a Gist  https://api.github.com/gi...  ...Tful API / Test Suite / Delete a Gist  204 No Content  124 ms  0
```

Reference

https://www.vinaysahni.com/best-practices-for-a-pragmatic-restful-api

https://hackernoon.com/restful-api-designing-guidelines-the-best-practices-60e1d954e7c9

http://www.restapitutorial.com/httpstatuscodes.html

https://www.getpostman.com/docs/v6/

http://blog.getpostman.com/2014/03/07/writing-automated-tests-for-apis-using-postman/

https://jwt.io/