

Postman – wykonywanie żądań do API

Początek laboratorium:

- zainstalować program *Postman*: <https://www.postman.com/downloads/>,
- pominąć zakładanie konta: *Skip and go to the app/lightweight API client*.

Create a free Postman account

Work email

Create Free Account

Already have an account? [Log In](#)

Or continue with the [lightweight API client](#).

Zadania (*Postman*, *API NBP*):

Zadanie 10.1:

Wyjaśnić następujące zagadnienia:

- *API (Application Programming Interfaces)*,
- *WebAPI (API webowe/API internetowe/interfejsy webowe)*,
- *publiczne a prywatne API*,
- *ograniczenie liczby żądań (rate limit)*,
- *dokumentacja API*,
- *formaty danych wykorzystywane na potrzeby wymiany danych z API*,
- (przypomnienie) *protokół HTTP*, żądania/odpowiedzi: nagłówki, ciała,
- (przypomnienie) *przekazywanie parametrów w ścieżce URL (Path parameters)*,
- (przypomnienie) *przekazywanie parametrów w query string'u (Query parameters)*,
- *grupy statusów (kodów) HTTP*, ich rola, omówić każdą z nich,
- (przypomnienie) *metody HTTP: GET i POST*,
- *inne metody HTTP: PUT, DELETE, PATCH, HEAD, OPTIONS*,
- *jeszcze inne metody HTTP: CONNECT, TRACE, **
- *nagłówek żądania Accept i jego wartości: application/...*
- *nagłówek żądania/odpowiedzi Content-Type i jego wartości: application/...*

–

Zadanie 10.2:

Zapoznać się z API Narodowego Banku Polskiego.

Uruchomić program *Postman*.

Wykonać żądania *HTTP* w celu uzyskania informacji o tabeli kursów (domyślnie w formacie *JSON*). Zwrócić uwagę, kiedy żądanie zakończyło się „sukcesem” (statusy *2XX*), a kiedy niepowodzeniem (statusy *4XX*):

- typu A,
- typu A ustalając w ciągu zapytania zwracanie odpowiedzi w formacie XML,
- typu A ustalając w nagłówku zwracanie odpowiedzi w formacie XML,
- typu B,
- typu D,
- typu A z trzech ostatnich dostępnych dni,
- typu A z dziś,
- typu A z 2 maja 2024,
- typu A z 3 maja 2024,
- typu A z dostępnych dni pomiędzy 29 kwi. 2024 a 5 maja 2024,
- typu A z dostępnych dni pomiędzy 1 sty. 2024 a 5 maja 2024,
- typu A z dostępnych dni pomiędzy 1 sty. 2001 a 2 lut. 2001.

<http://api.nbp.pl/>

Status: 400 Bad Request - Nieprawidłowa wartość parametru: {t:

Status: 404 Not Found - Brak danych Time: 71 ms

400 Bad Request - Nieprawidłowa wartość parametru: {table}='D'

404 Not Found - Brak danych

ość par

The request cannot be fulfilled due to bad syntax.

The requested resource could not be found but may be available again in the future. Subsequent requests by the client are permissible.

Zadanie 10.3: *

Wykonać kolejne żądania *HTTP* w celu uzyskania informacji o kursie pojedynczej waluty (domyślnie w formacie *JSON*):

- *euro* z tabeli A,
- *euro* z tabeli B,
- *funta* z tabeli A,
- *funta* z tabeli B,
- 7 ostatnich kursów *euro* z tabeli A,
- 256 ostatnich kursów *euro* z tabeli A,
- *euro* z tabeli A z dzisiaj,
- *euro* z tabeli A z 2 maja 2024,
- *euro* z tabeli A z 3 maja 2024,
- *dolara ameryk.* z dostępnych dni pomiędzy 29 kwi. 2024 a 5 maja 2024,
- *dolara ameryk.* z dostępnych dni pomiędzy 1 sty. 2023 a 5 maja 2024,
- *dolara ameryk.* z dostępnych dni pomiędzy 1 sty. 2001 a 2 lut. 2001.

```
1  {
2    "table": "A",
3    "currency": "euro",
4    "code": "EUR",
5    "rates": [
6      {
7        "no": "086/A/NBP/2024",
8        "effectiveDate": "2024-05-02",
9        "mid": 4.3323
10     }
11   ]
12 }
```

Zadanie 10.4:

Wykonać kolejne żądania *HTTP* w celu uzyskania informacji o cenie złota (domyślnie w formacie *JSON*):

- aktualna cena złota,
- ostatnich 7 notowań ceny złota,
- ostatnich -7 notowań ceny złota,
- ostatnich 256 notowań ceny złota,
- cena złota z 2 maja 2024,
- cena złota z 3 maja 2024,
- cena złota z dostępnych dni pomiędzy 29 kwi. 2024 a 5 maja 2024.

```
1 [
2   {
3     "data": "2024-05-02",
4     "cena": 298.60
5   }
6 ]
```

Zadania (*Postman*, *API GitHub'a*):

Zadanie 10.5:

Zapoznać się z *API GitHub'a*.

Wykonać żądanie *GET* sprawdzające limit żądań (sprawdzić, czy jest mniej niż 60):

- https://api.github.com/rate_limit.

W przypadku wyczerpania limitu żądań przejść do zadania 10.10 w celu wygenerowania *PATokena*, którego użycie znacznie zwiększa limit.

Używać *token* jak w zadaniu 10.11.

<https://docs.github.com/en/rest?apiVersion=2022-11-28>

GitHub REST API documentation

Create integrations, retrieve data, and automate your workflows with the GitHub REST API.

Overview

Quickstart

<https://docs.github.com/en/rest/overview/rate-limits-for-the-rest-api?apiVersion=2022-11-28>

<https://docs.github.com/en/rest/rate-limit/rate-limit?apiVersion=2022-11-28>

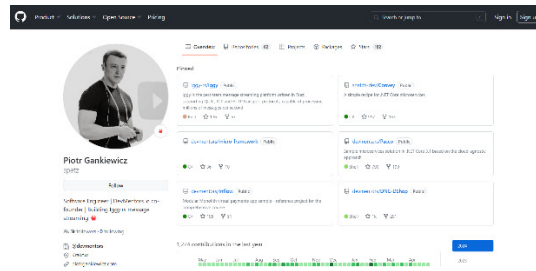
The screenshot shows a Postman interface with a GET request to `https://api.github.com/rate_limit`. The response is a JSON object with the following structure:

```
{
  "resources": {
    "core": {
      "limit": 60,
      "remaining": 60,
      "reset": 1714827883,
      "used": 0,
      "resource": "core"
    }
  }
}
```

The status bar indicates a 200 OK response with a time of 44 ms and a size of 1.32 KB.

Zadanie 10.6:

W przeglądarce internetowej przejść na profil użytkownika *spetz* (Software Engineer z Krakowa, autor kilku tutoriali): <https://github.com/spetz>



Zadanie 10.7:

Powrócić do Postman'a, w którym wykonać żądania GET pod następujące adresy:

- <https://api.github.com/users>
- <https://api.github.com/users?since=2419599>
- <https://api.github.com/users/spetz>
- <https://api.github.com/users/spetz/hovercard>
- <https://api.github.com/users/spetz/followers>
- <https://api.github.com/users/spetz/followers?page=5>
- <https://api.github.com/repos/spetz/tokenmanager-sample>

<https://docs.github.com/en/rest/users/users#get-a-user>

<https://docs.github.com/en/rest/users/users#get-contextual-information-for-a-user>

<https://docs.github.com/en/rest/users/followers#list-followers-of-a-user>

<https://docs.github.com/en/rest/repos/repos?apiVersion=2022-11-28#get-a-repository>

Zadanie 10.8:

Wykonać żądania GET w celu uzyskania informacji/pobrania:

- o szablonie pliku `.gitignore` („surową” zawartość pliku) dla projektów *Laravel'a*,
- o (nieistniejące) repozytorium *TokenmanagerSample* użytkownika *spetz*,
- dane o pliku *Readme.md* i jego zawartość zakodowaną z repozytorium *tokenmanager-sample* użytkownika *spetz*,
- „surowy” zawartość pliku *README.md* z repozytorium *tokenmanager-sample* użytkownika *spetz*,
- współtwórców repozytorium *Passenger* użytkownika *passenger-stack*,
- czy użytkownik *phuzarski* jest współtwórcą repozytorium *Passenger* użytkownika *passenger-stack*,
- czy użytkownik *phuzarski* follow'uje *spetz*,
- czy użytkownik *spetz* follow'uje *phuzarski*,
- repozytorium *Passenger* użytkownika *passenger-stack* jako archiwum `.zip`,
- liczbę klonowań repozytorium *Passenger* użytkownika *passenger-stack*,
- informacje o *commit'ach* z repozytorium *Passenger* użytkownika *passenger-stack*,
- informacje o *commicie* o *hash'u*: `b1c3a8011e585eba20d85ba3cdbc9f5dbe29f345` z repozytorium *Passenger* użytkownika *passenger-stack*,
- informacje o *gałęziach* z repozytorium *Passenger* użytkownika *passenger-stack*,
- informacje o gałęzi *master* z repozytorium *Passenger* użytkownika *passenger-stack*,
- informacje o gałęzi *main* z repozytorium *Passenger* użytkownika *passenger-stack*.

<https://docs.github.com/en/rest/gitignore/gitignore?apiVersion=2022-11-28#get-a-gitignore-template>

<https://docs.github.com/en/rest/repos/repos?apiVersion=2022-11-28#get-a-repository>

<https://docs.github.com/en/rest/repos/contents?apiVersion=2022-11-28#get-repository-content>

<https://docs.github.com/en/rest/repos/contents?apiVersion=2022-11-28#about-repository-contents>

<https://docs.github.com/en/rest/repos/repos?apiVersion=2022-11-28#list-repository-contributors>

<https://docs.github.com/en/rest/collaborators/collaborators?apiVersion=2022-11-28#check-if-a-user-is-a-repository-collaborator>

<https://docs.github.com/en/rest/users/followers#check-if-a-user-follows-another-user>

<https://docs.github.com/en/rest/repos/contents?apiVersion=2022-11-28#download-a-repository-archive-zip>

<https://docs.github.com/en/rest/metrics/traffic?apiVersion=2022-11-28#get-repository-clones>

<https://docs.github.com/en/rest/commits/commits?apiVersion=2022-11-28#list-commits>

<https://docs.github.com/en/rest/commits/commits?apiVersion=2022-11-28#get-a-commit>

<https://docs.github.com/en/rest/git/commits?apiVersion=2022-11-28#get-a-commit-object>

<https://docs.github.com/en/rest/branches/branches?apiVersion=2022-11-28#list-branches>

<https://docs.github.com/en/rest/branches/branches?apiVersion=2022-11-28#get-a-branch>

Zadanie 10.9:

Wykonać żądanie *POST*, dzięki któremu jest możliwość:

- wyrenderowania fragmentu *Markdown* (dołączonego jako *JSON* w ciele żądania) jako *HTML* (zakładka *Preview* w odpowiedzi).

```
{"text": "# h1 Heading \n ## h2 Heading\n### h3 Heading\n#### h4 Heading\n##### h5 Heading"}
```

<https://docs.github.com/en/rest/markdown/markdown?apiVersion=2022-11-28>

POST ▼ https://api.github.com/markdown

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▼

1 `{"text": "# h1 Heading \n ## h2 Heading\n### h3 Heading\n#### h4 Heading\n##### h5 Heading"}`

Body Cookies Headers (26) Test Results 🌐 Status: 200 OK

Pretty Raw **Preview** Visualize

h1 Heading

h2 Heading

h3 Heading

h4 Heading

h5 Heading

Zadania (Postman, API GitHub'a, autoryzacja):

Uwaga: żądania POST, PUT, PATCH, DELETE mogą spowodować modyfikację/utratę danych użytkownika!

Zadanie 10.10:

Zalogować się na swoje konto na *GitHub*'ie oraz wygenerować klasyczny *token* osobisty: dowolna nazwa, ważny 1 dzień.

Udzielić uprawnienia do: *repo*, *user*, *delete_repo*, *project*.

Skopiować go tymczasowo do pliku tekstowego w *Notepad'zie++*.

Po skończonych zajęciach dla bezpieczeństwa usunąć *token*.

<https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens>

Settings → Developer Settings → Personal access tokens → Tokens (classic) → Generate new token

(lub przejść pod <https://github.com/settings/personal-access-tokens/new>)

Personal access tokens (classic) Generate new token ▼

Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the [GitHub API](#).

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Generate new token ▼

- Generate new token (Beta)**
Fine-grained, repo-scoped
- Generate new token (classic)**
For general use

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> user	Update ALL user data
<input checked="" type="checkbox"/> read:user	Read ALL user profile data
<input checked="" type="checkbox"/> user:email	Access user email addresses (read-only)
<input checked="" type="checkbox"/> user:follow	Follow and unfollow users
<input checked="" type="checkbox"/> delete_repo	Delete repositories
<input checked="" type="checkbox"/> project	Full control of projects
<input checked="" type="checkbox"/> read:project	Read access of projects

Zadanie 10.11:

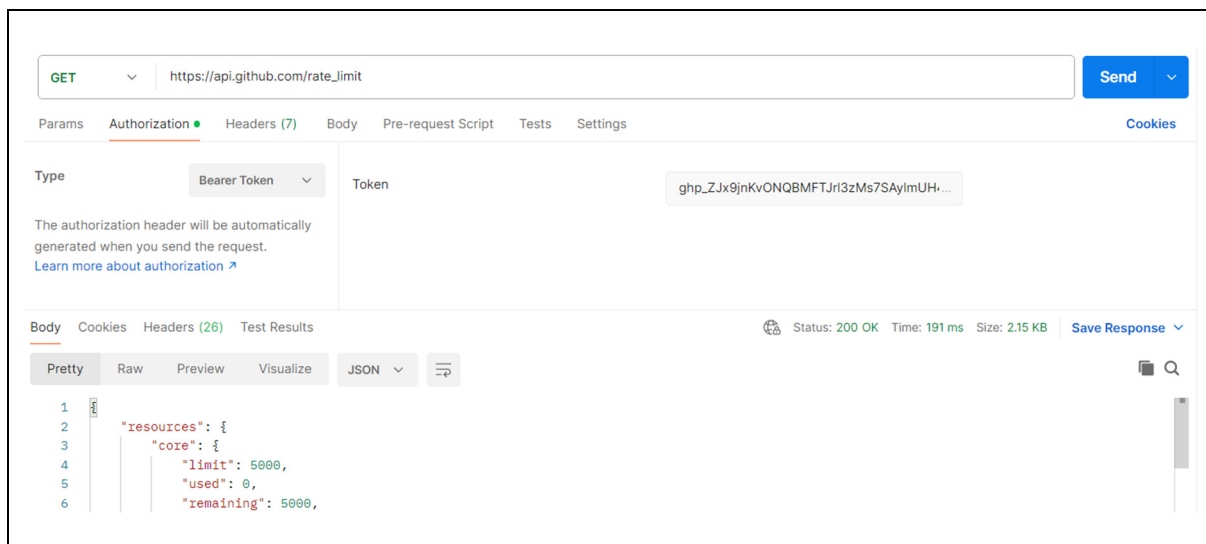
Wykonać żądanie GET sprawdzające limit żądań (sprawdzić, czy jest 5000):

• https://api.github.com/rate_limit.

wykorzystując *PAToken* do uwierzytelnienia.

Authorization -> Bearer Token -> wkleić *PAToken*

<https://docs.github.com/en/rest/authentication/authenticating-to-the-rest-api#authenticating-with-a-personal-access-token>



GET https://api.github.com/rate_limit Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Type Bearer Token Token ghp_ZJx9jnKvONQBMFTJrI3zMs7SAyimUH...

The authorization header will be automatically generated when you send the request.
[Learn more about authorization](#)

Body Cookies Headers (26) Test Results Status: 200 OK Time: 191 ms Size: 2.15 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "resources": {
3     "core": {
4       "limit": 5000,
5       "used": 0,
6       "remaining": 5000,
```

Zadanie 10.12:

Wykorzystać **PAToken** do uwierzytelnienia się w żądaniach (zachować ostrożność):

- uzyskania informacji o sobie,
- zaktualizowania jakiejś informacji o sobie np. lokalizacja na PL,
- uzyskania informacji o swoich wszystkich repozytoriach,
- uzyskania informacji o swoich *prywatnych* repozytoriach,
- uzyskania informacji o swoich *publicznych* repozytoriach,
- utworzenia nowego prywatnego repozytorium:
 - nazwa → ai1-test,
 - opis → *This is AI1 Test repository!*,
 - prywatne → tak.
- przestawić (zaktualizować częściowo) przed chwilą utworzone repozytorium na publiczne,
- usunąć całkiem to repozytorium,
- spróbować jeszcze raz usunąć to repozytorium.

Po wykonaniu każdego żądania, sprawdzać/porównywać rezultat także poprzez interfejs użytkownika na github.com.

<https://docs.github.com/en/rest/users/users?apiVersion=2022-11-28#get-the-authenticated-user>

<https://docs.github.com/en/rest/users/users?apiVersion=2022-11-28#update-the-authenticated-user>

<https://docs.github.com/en/rest/repos/repos?apiVersion=2022-11-28#list-repositories-for-the-authenticated-user>

<https://docs.github.com/en/rest/repos/repos?apiVersion=2022-11-28#create-a-repository-for-the-authenticated-user>

<https://docs.github.com/en/rest/repos/repos?apiVersion=2022-11-28#update-a-repository>

<https://docs.github.com/en/rest/repos/repos?apiVersion=2022-11-28#delete-a-repository>

Zadanie 10.13: *

Przed następnymi zajęciami zapoznać się z następującymi zagadnieniami:

- **REST** (*Representational state transfer*),
- **REST API** (*RESTful API*),
- 6 zasad **REST API** (w tym wyjaśnić dokładnie zasadę *bezstanowości*),
- *endpoint'y*,
- *zasoby (resources)*,
- operacje **CRUD** na zasobach,

- zagadnienie *modelu dojrzałości API Richardsona*, omówić poziomy,
- *metody HTTP* używane na 2 poziomie dojrzałości,
- *kody HTTP* (nazwa angielska, tłumaczenie, najczęstsze zastosowanie):
200, 201, 204, 400, 401, 403, 404, 405, 422, 500,
- ocenić poziom dojrzałości API *GitHub'a*,
- wersjonowanie API,
- czy API *GitHub'a* jest wersjonowane?

<https://zaprogramujzycie.pl/czym-jest-rest/>

<https://www.redhat.com/en/topics/api/what-is-a-rest-api>

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

<https://www.webfx.com/web-development/glossary/http-status-codes/>

<https://web.archive.org/web/20220519043646/https://jakzostacprogramista.net/2020/11/25/rest-api-model-dojrzalosci-richardsona/>

★ – zadania/podpunkty do samodzielnego dokończenia/wykonania,

★ – zadania/podpunkty dla zainteresowanych.

Po zakończonym laboratorium należy skasować wszystkie pobrane oraz utworzone przez siebie pliki z komputera w sali laboratoryjnej.

Wersja pliku: v1.01