

Laboratorium 4

Wykorzystanie REST API z narzędzia Postman oraz frameworka Express.js

Na stronie <http://rperlinski.herokuapp.com/api> jest proste API pozwalające na operacje CRUD. We wszystkich zadaniach z wyjątkiem pierwszego można skorzystać z pakietu [request](#).

Zadanie 1 – użycie narzędzia do obsługi REST

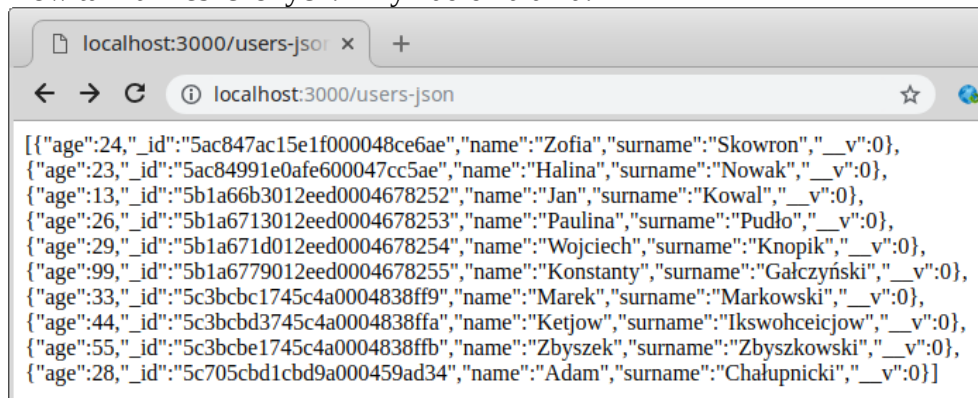
Korzystając z narzędzia do pracy z REST (np. [Postman](#) dla Chrome) przetestować działanie API, wszystkie cztery operacje:

- pobranie listy użytkowników (GET) i pojedynczego użytkownika podając jego id (GET),
- utworzenie nowego użytkownika (POST),
- aktualizacja danych istniejącego użytkownika podając jego id (PUT),
- usunięcie jednego użytkownika podając jego id (DELETE).

Wykorzystanie REST API z poziomu frameworka Express.js

Zadanie 2 – pobranie listy użytkowników z REST API za pomocą pakietu [request](#)

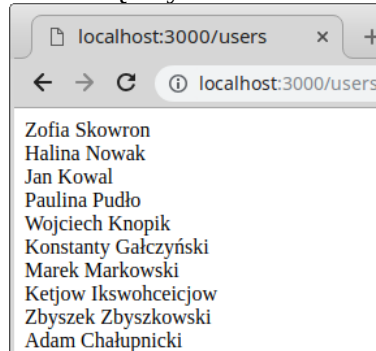
Pod adresem `/users-json` proszę stworzyć obsługę żądania do zasobów API <http://rperlinski.herokuapp.com/api/users> i wyświetlić w formacie JSON tablicę wszystkich użytkowników tam umieszczonych. Przykład działania:



```
[{"age":24,"_id":"5ac847ac15e1f00048ce6ae","name":"Zofia","surname":"Skowron","_v":0}, {"age":23,"_id":"5ac84991e0afe600047cc5ae","name":"Halina","surname":"Nowak","_v":0}, {"age":13,"_id":"5b1a66b3012eed0004678252","name":"Jan","surname":"Kowal","_v":0}, {"age":26,"_id":"5b1a6713012eed0004678253","name":"Paulina","surname":"Pudło","_v":0}, {"age":29,"_id":"5b1a671d012eed0004678254","name":"Wojciech","surname":"Knopik","_v":0}, {"age":99,"_id":"5b1a6779012eed0004678255","name":"Konstanty","surname":"Galczyński","_v":0}, {"age":33,"_id":"5c3bcbc1745c4a0004838ff9","name":"Marek","surname":"Markowski","_v":0}, {"age":44,"_id":"5c3bcbd3745c4a0004838ffa","name":"Ketjow","surname":"Ikswohceicjow","_v":0}, {"age":55,"_id":"5c3bcbe1745c4a0004838ffb","name":"Zbyszek","surname":"Zbyszkowski","_v":0}, {"age":28,"_id":"5c705cbd1cbd9a000459ad34","name":"Adam","surname":"Chałupnicki","_v":0}]
```

Zadanie 3 – sformatowana lista użytkowników

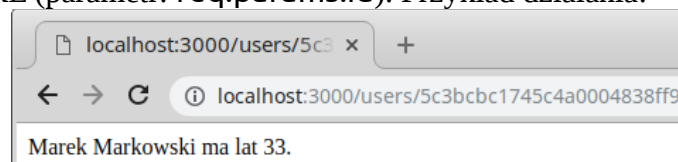
Pod adresem `/users` proszę wyświetlić listę użytkowników zawierającą imię i nazwisko każdego z nich. Przykład działania:



```
Zofia Skowron  
Halina Nowak  
Jan Kowal  
Paulina Pudło  
Wojciech Knopik  
Konstanty Galczyński  
Marek Markowski  
Ketjow Ikswohceicjow  
Zbyszek Zbyszkowski  
Adam Chałupnicki
```

Zadanie 4 – dane konkretnego użytkownika

Pod adresem `/users/id` proszę wyświetlić informację o konkretnym użytkowniku, którego ID podano w adresie URL (parametr: `req.params.id`). Przykład działania:

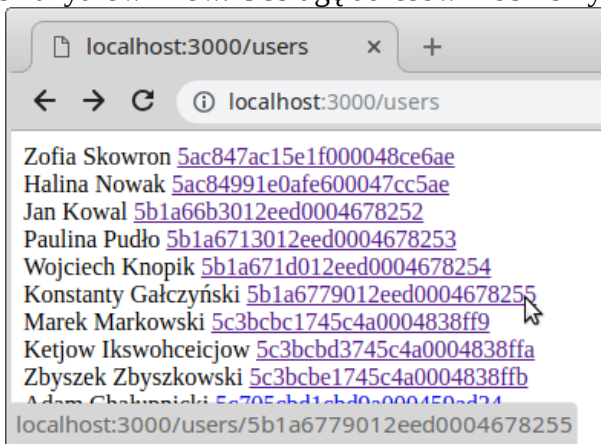


```
localhost:3000/users/5c3bcbd3745c4a0004838ff9  
Marek Markowski ma lat 33.
```

Zadanie 5 – lista użytkowników z adresem URL konkretnego użytkownika

Modyfikujemy kod z zadania 3 aby oprócz imienia i nazwiska zawierał jeszcze odnośnik do adresów konkretnych użytkowników. Obsługę adresów zrobiliśmy w zadaniu 4.

Przykład działania:



Zadanie 6 – inne zabawy dla chętnych

Proszę dodać do strony przycisk, którego kliknięcie wyliczy i wyświetli statystyki imion występujących w użytkownikach dostępnych poprzez API. Chodzi o listę imion z liczbą ich wystąpień, np.:

- Zofia występuje 1 razy.
- Robert występuje 2 razy.
- Mariusz występuje 3 razy.
- Marek występuje 4 razy.
- Hello występuje 1 razy.
- Jolka występuje 1 razy.

Zadanie 7 – możliwość dodawania i aktualizacji danych poprzez formularze

Stworzyć prostą stronę HTML pozwalającą na zarządzanie danymi z wykorzystaniem podanego API. To zadanie można sobie rozbić na części:

- dodać formularz dodawania użytkowników:** kontrolki name, surname, age i przycisk oraz obsługa przesłania danych do API wykorzystując AJAX, metodę POST,
- dodać formularz aktualizacji,** który po kliknięciu w przycisk edycji wypełni się danymi wybranej osoby; wysłanie formularza aktualizacji aktualizuje dane osoby poprzez API,
- dodać przycisk/odnośnik usuwania wybranego użytkownika,** po kliknięciu którego użytkownik zostanie usunięty korzystając z metody DELETE protokołu HTTP.