

Національний Технічний університет України

“Київський політехнічний інститут”

імені Ігоря Сікорського

Лабораторна робота №5

з дисципліни "**ОСНОВИ ХМАРНИХ ОБЧИСЛЕНЬ**"

**Створення Application  
Programming Interface для хмарного додатку**

Виконав:

студент групи Да-01мн

Терещенко О.І

**Мета роботи:** створити простий HTTP API для хмарного додатку з лабораторної роботи № 4. API має задовольняти REST вимогам побудови інтерфейсів. Продемонструвати роботу API на прикладі 3-4 методів. Кожен метод перевірити з клієнту за допомогою curl або інших аналогічних програм.

**Задача:** використати код програми з лабораторної роботи № 4 та побудувати приклад 3-4 методів API, які будуть задовольняти REST вимогам. Для передачі даних у методах використовувати JSON або XML формат.

**Короткі теоретичні відомості**

REST (скор. англ. Representational State Transfer, «передача стану подання») — підхід до архітектури мережевих протоколів, які забезпечують доступ до інформаційних ресурсів. Був описаний і популяризований у 2000 році Роєм Філдінгом (Roy Fielding), одним із творців протоколу HTTP. Найвідомішою системою, побудованою переважно за архітектурою REST, є сучасна Всесвітня павутина.

Дані повинні передаватися у вигляді невеликої кількості стандартних форматів (наприклад HTML, XML, JSON). Мережевий протокол (як і HTTP) повинен підтримувати кешування, не повинен залежати від мережевого прошарку, не повинен зберігати інформацію про стан між парами «запит-відповідь». Стверджується, що такий підхід забезпечує масштабування системи і дозволяє їй еволюціонувати з новими вимогами.

Антиподом REST є підхід, заснований на виклику віддалених процедур (Remote Procedure Call, RPC). Підхід RPC дозволяє використовувати невелику кількість мережевих ресурсів з великою кількістю методів і складним протоколом. При підході REST кількість методів і складність протоколу суворо обмежені, що призводить до того, що кількість окремих ресурсів має бути великою.

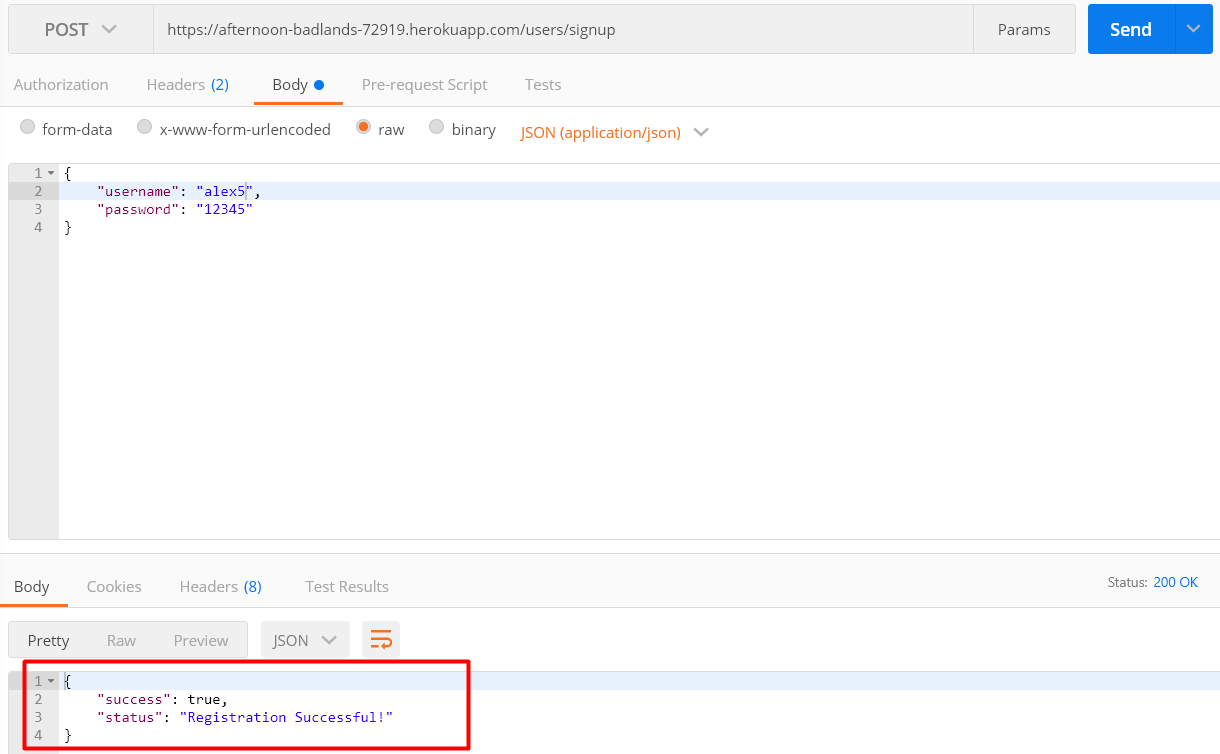
JSON (англ. JavaScript Object Notation, укр. об'єктний запис JavaScript, вимовляється джейсон) — це легкий формат обміну даними між комп'ютерами. JSON базується на тексті, і може бути з легкістю прочитаним людиною. Формат дозволяє описувати об'єкти та інші структури даних. Цей формат головним чином використовується для передачі структурованої інформації через мережу (завдяки процесу, що називають серіалізацією). Розробив і популяризував формат Дуглас Крокфорд.

JSON знайшов своє головне призначення у написанні веб-програм, а саме при використанні технології AJAX. JSON виступає як заміна XML під час асинхронної передачі структурованої інформації між клієнтом та сервером. При цьому перевагою JSON перед XML є те, що він дозволяє складні структури в атрибутах, займає менше місця і прямо інтерпретується за допомогою JavaScript в об'єкти.

**Варіант згідно додатку 1:**



**Хід роботи:**

Для надсилання запитів використаємо програму Postman. Посилаємо POST запит створення нового користувача: 

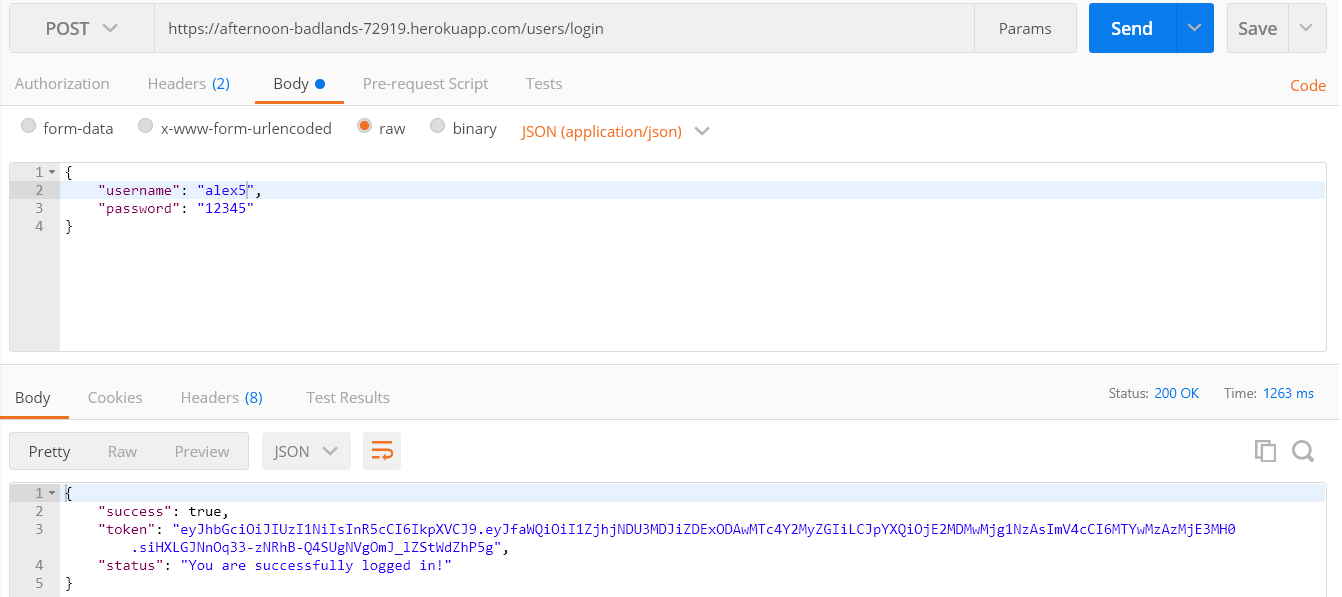
В MongoDB Atlas в БД було додано нового користувача:



Код обробки запиту на сервері:



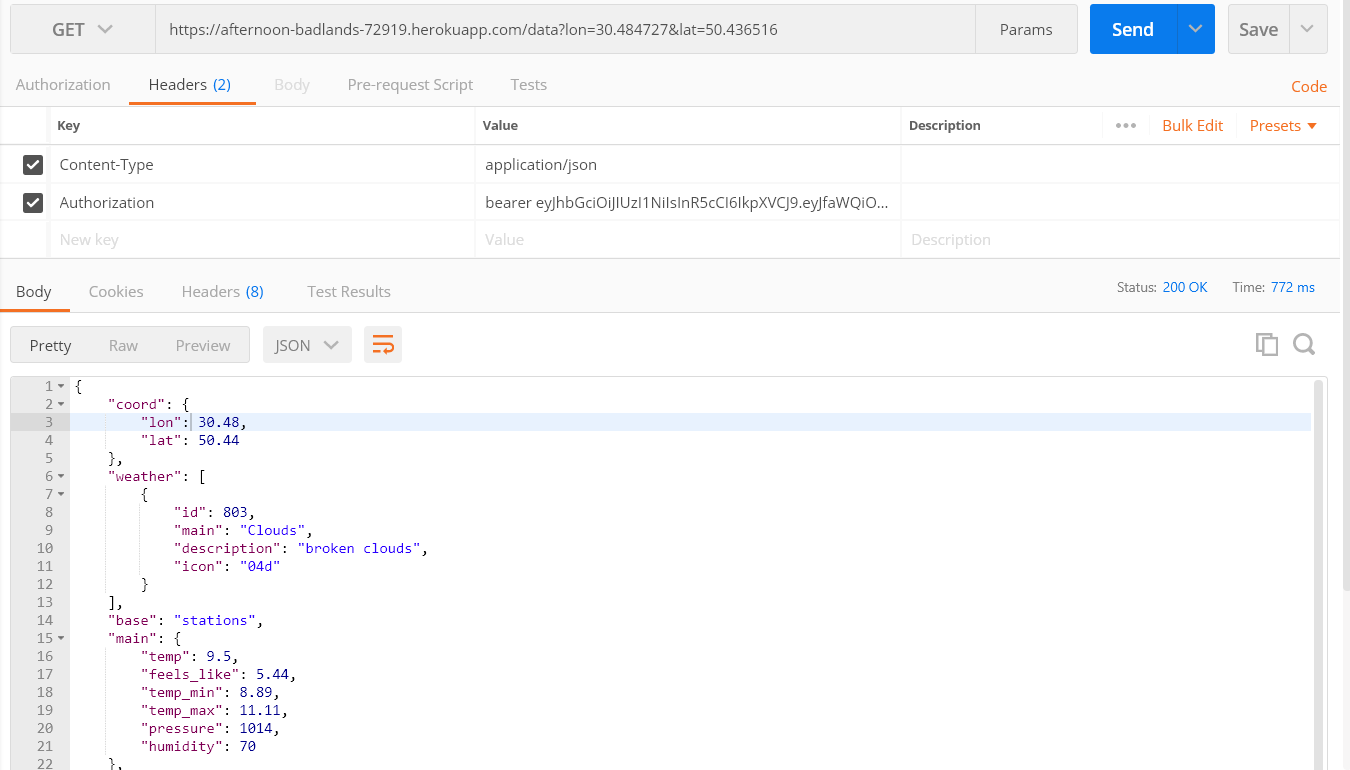
Надсилаємо запит на авторизації у додатку. Отримуємо JWT токен, який будемо надсилати при наступних запитах у заголовку Authorization.



Код, що відповідає за обробку запиту авторизації:



Отримуємо дані про погоду за координатою:



<https://afternoon-badlands-72919.herokuapp.com/data?lon=30.484727&lat=50.436516>

Формат отриманих даних:

{

"coord": {

"lon": 30.48,

"lat": 50.44

},

"weather": [

{

"id": 803,

"main": "Clouds",

"description": "broken clouds",

"icon": "04d"

}

],

"base": "stations",

"main": {

"temp": 9.5,

"feels\_like": 5.44,

"temp\_min": 8.89,

"temp\_max": 11.11,

"pressure": 1014,

"humidity": 70

},

"visibility": 10000,

"wind": {

"speed": 4,

"deg": 250

},

"clouds": {

"all": 75

},

"dt": 1603029440,

"sys": {

"type": 1,

"id": 8903,

"country": "UA",

"sunrise": 1602995138,

"sunset": 1603033223

},

"timezone": 10800,

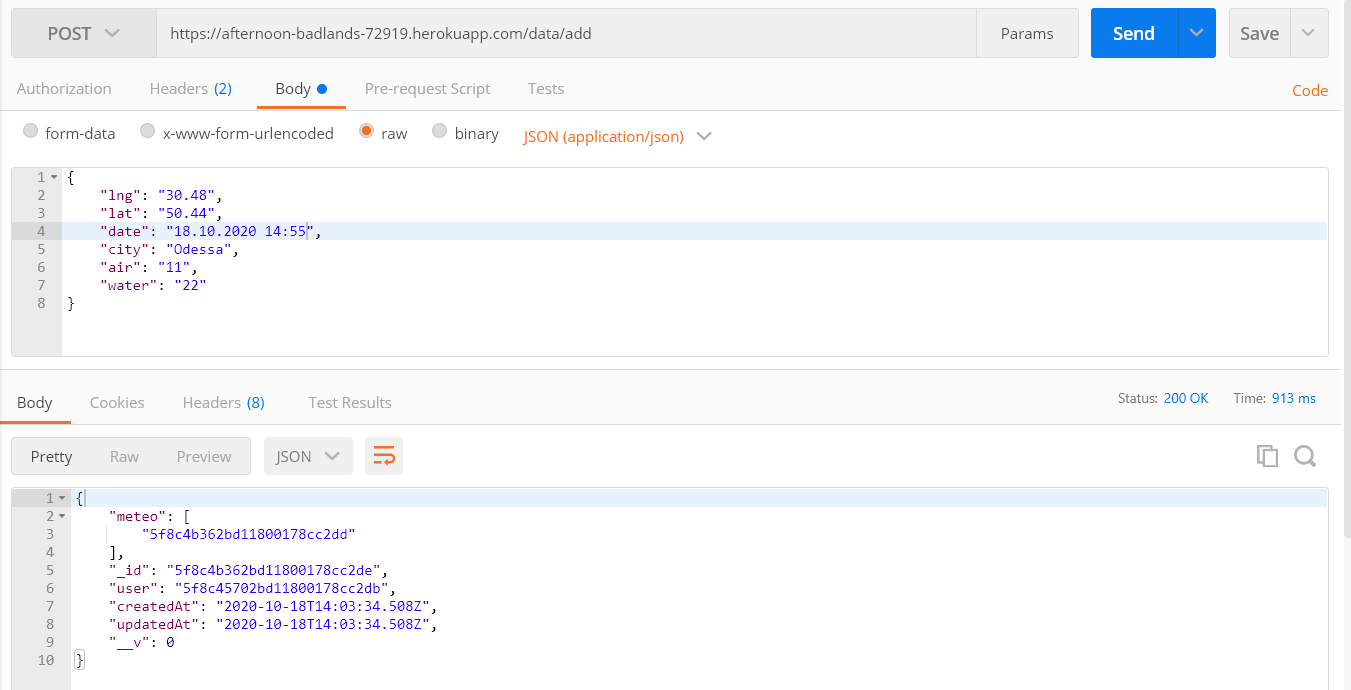
"id": 696050,

"name": "Pushcha-Vodytsya",

"cod": 200

}

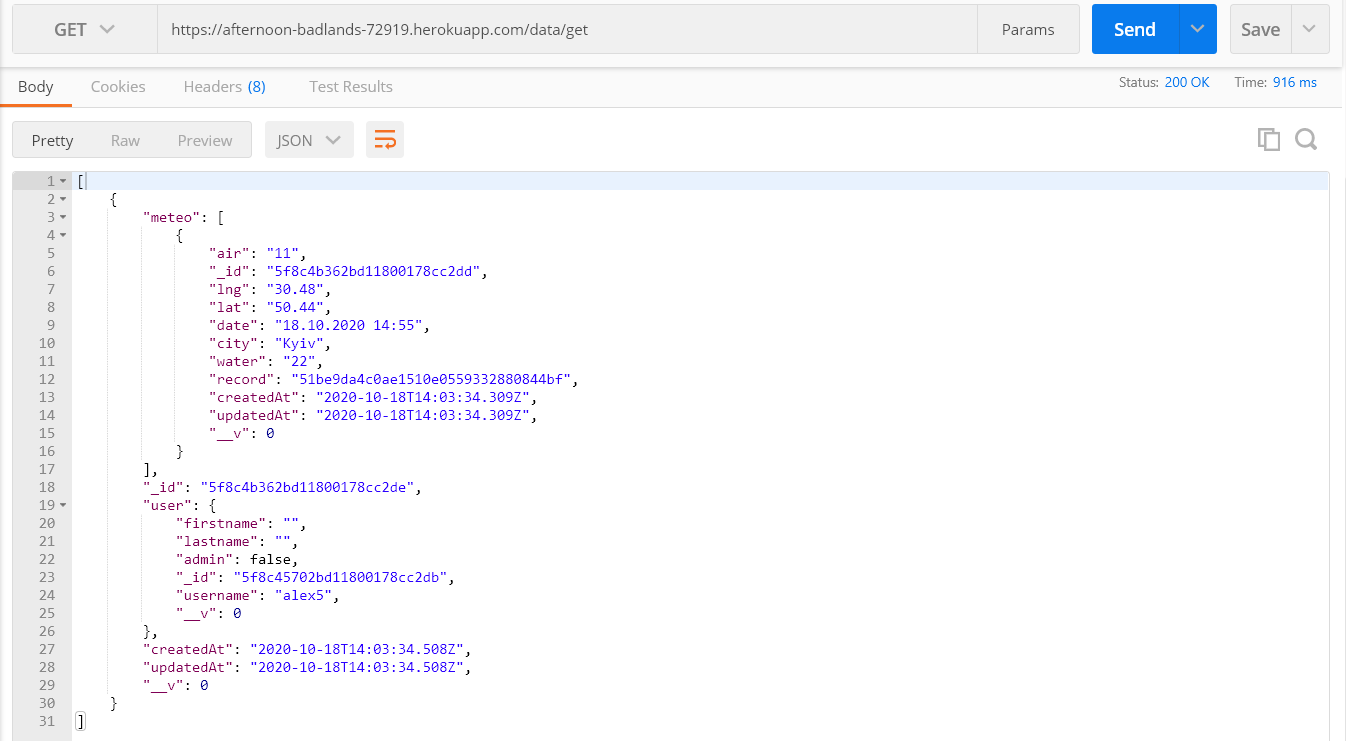
Додаємо дані про температуру до нашої БД:



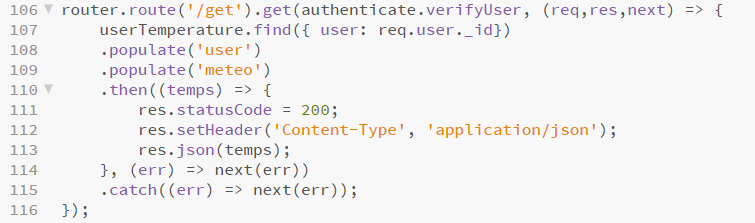
Код, що обробляє цей запит:



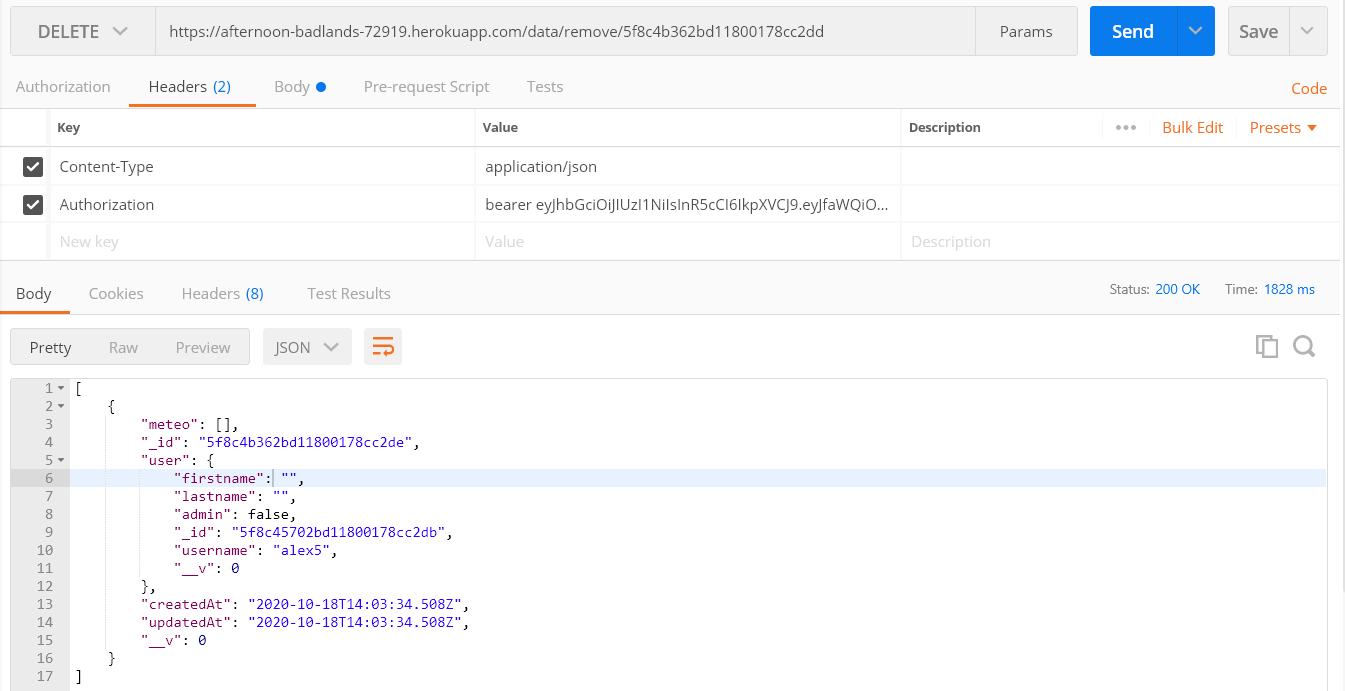
Отримуємо список створених записів про температуру:



Код, що обробляє цей запит:



Видалення запису про температуру з БД. В результаті сервер повертає оновлений список записів користувача:



Код:



**Висновок:**

В лабораторній роботі було створено HTTP API для хмарного додатку, розробленого у минулій лабораторній роботі. Форматом отримання даних було обрано JSON, так як він легко конвертується у JS об’єкти. Було реалізована підтримка отримання даних про погоду, додавання нового користувача, додавання, видалення записів користувачів і т.д. Після розміщення додатку на Heroku, було протестовано роботу таких запитів до додатку.