

УДК 004.4:502

# **СТВОРЕННЯ ВЕБ-ДОДАТКУ ДЛЯ ОЗНАЙОМЛЕННЯ З РАДІАЦІЙНИМ ТА ЕКОЛОГІЧНИМ СТАНОМ В УКРАЇНІ НА ОСНОВІ ДАНИХ З OPENDATA ЗА ДОПОМОГОЮ ВИКОРИСТАННЯ ТЕХНОЛОГІЇ GRAPHQL**

**Сокол Марина**

**Науковий керівник: завідувач кафедри**

**доцент, кандидат фізико-математичних наук Парашук С. Д.**

*Центральноукраїнський державний педагогічний університет імені Володимира  
Винниченка, м. Кропивницький, Україна*

*Статтю присвячено проблемам екології довкілля України, а саме розробці відповідного додатку, який на основі даних з OpenData ознайомлює потенційних користувачів з радіаційним та екологічним станом в Україні. Для створення додатку використано технологію GraphQL, що дозволяє отримувати лише потрібні дані від сервера, що дозволяє знизити витрати трафіку та часу користувача. Також GraphQL мінімізує дані, отже розробник отримує лише той тип даних, що має описано в документації. Представлена у статті система надає змогу стежити за даними екологічного стану довкілля, які надає держава, в найбільш актуальному стані, адже не потребує змін при оновленні даних і працює цілком автономно.*

*Ключові слова: open data, екологія, радіація, мапа, graphql.*

**Creating a web application to familiarize yourself with radiation and  
environmental conditions in Ukraine based on OpenData data using GraphQL  
technology**

**Sokol Maryna**

**Scientific adviser: Head of the department**

**Assistant Professor, Candidate of Physical and Mathematical Sciences**

**Paraschuk S.D.**

*Volodymyr Vynnychenko Central Ukrainian State Pedagogical University,  
Kropyvnytskyi, Ukraine*

*The article is devoted to environmental problems of Ukraine, namely the development of a suitable application that, based on data from OpenData, introduces potential users to the radiation and environmental situation in Ukraine. GraphQL technology is used to create the application, which allows to receive only the necessary data from the server, which reduces the traffic and user time. GraphQL also typifies the data, so the developer receives only the type of data that is described in the documentation. The system presented in the article makes it possible to monitor the state of the environment in the most up-to-date state of the environment, since it does not require any changes in data updates and works completely autonomously.*

*Keywords: open data, ecology, radiation, map, GraphQL.*

**Постановка проблеми:** Останні сто років розвитку людського суспільства характеризуються все більш інтенсивним і прискореним зростанням за всіма показниками. Щоб задовольнити потреби всієї цієї маси людей ще швидшими темпами зростало промислове виробництво, а також використання хімічних засобів у сільському господарстві. Крім швидкого зростання чисельності населення спостерігається і інтенсивне зростання потреб людини. Виросли стандарти життя. Сучасне індустріальне суспільство можна характеризувати як суспільство споживання. Задовольняючи все більше нових потреб людини, виробництво непомітно і безповоротно при цьому знищувало природні ресурси, які необхідні для задоволення природних потреб людини [1].

Завдяки Закону України «Про доступ до публічної інформації» та постанови Кабінету Міністрів України від 21 жовтня 2015 р. № 835 «Про затвердження Положення про набори даних, які підлягають оприлюдненню у формі відкритих даних» у нашій країні з'явився єдиний державний веб-портал відкритих даних, де кожен може вільно і безкоштовно ознайомитися з інформацією, що надають органи влади. Не зважаючи на це, вигляд цієї інформації найчастіше не є зручним та інформативним для звичайних користувачів та для розробників, що використовують ці дані для своїх проєктів (рис.1).

Тип даних	Формат даних
Текстові дані	TXT, RTF, ODT*, DOC(X), PDF (з текстовим змістом, нескановане зображення), (X)HTML*
Структуровані дані	RDF*, XML*, JSON*, CSV*, XLS(X), ODS*, YAML*
Графічні дані	GIF*, TIFF, JPG (JPEG)*, PNG*
Відеодані	MPEG, MKV, AVI, FLV, MKS, MK3D
Аудіодані	MP3, WAV, MKA
Дані, розроблені з використанням програми Macromedia Flash	SWF, FLV
Архів даних	ZIP*, 7z*, Gzip*, Bzip2*
Геопросторові дані	GeoTIFF, SHP, DMF, MID/MIF, DXF, XML, GeoJSON, GPX, LOC, ARINC, AIXM.

Рис.1 *Типи даних, які надає веб-портал*

Зазвичай це просто таблиці, опис яких потрібно шукати в іншому файлі, або не дуже інформативні списки. Тож, хоча держава й вільно надає всю важливу інформацію, але органи влади не переймаються тим, як цією інформацією будуть користуватися. На жаль, щоб зробити додаток з використанням інформації, що надає open data, розробник має вирішити низку проблем, таких як:

- відправлення великої кількості запитів
- отримання надлишкової інформації
- в результаті розробник отримує файл, з яким потрібно додатково попрацювати, щоб отримати зручне інформацію.

В той же час, в сучасному світі починають використовувати новий спосіб передачі даних — GraphQL, що є більш гнучким та краще підходить для проекту, де дані, їх кількість та вигляд не залежать від розробника [3], як при використанні даних з Open Data.

**Аналіз готових системи для відображення даних про екологічний стан довкілля.** Завдяки проекту Open Data Challenge розробники мають мотивацію для роботи з інформацією, що надає Open Data, тож є декілька готових проектів, але всі вони мають великі недоліки.

### 1) Зелений Кропивницький

Зелений Кропивницький [4] - це інструмент, що допомагає Міській раді міста Кропивницького інформувати громадськість про рішення, прийняті у сфері озеленення, а громадськості - контролювати ці рішення.

Першим недоліком сайту є занадто яскраві кольори першого екрану (Рис.2).



Рис. 2 Сайт «Зелений Кропивницький»

Також сайт не має адаптивності чи респонсивності, тобто з телефона чи планшета неможливо побачити корисну інформацію.

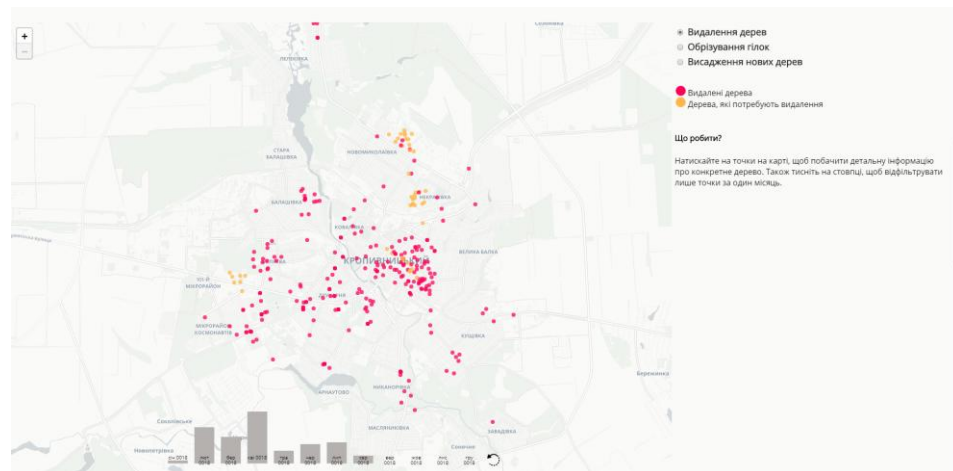


Рис. 4 Desktop версія карти сайту «Зелений Кропивницький»

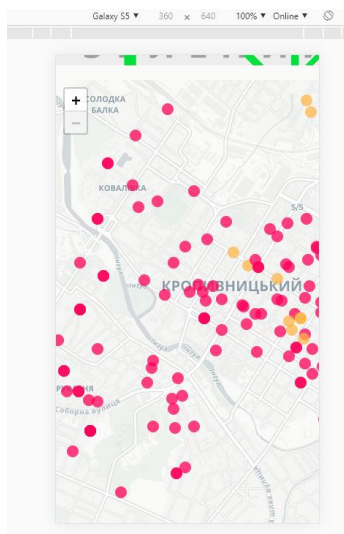


Рис. 5 Mobile версія карти сайту «Зелений Кропивницький»

Але основним мінусом є те, що сайт не показує нову інформацію, там є лише інформація за 2018 рік, тобто він не оновлюється і вже втратив свою актуальність.

## 2) Detox Ukraine

Detox Ukraine [1] – інтерактивна онлайн-карта об'єктів-забруднювачів довкілля з детальною інформацією по кожному джерелу викидів. Цей проект став фіналістом Open Data Challenge 2018. Виглядає гарно, але наразі зовсім не працює. Мобільна версія сайту також нормально не відображається.

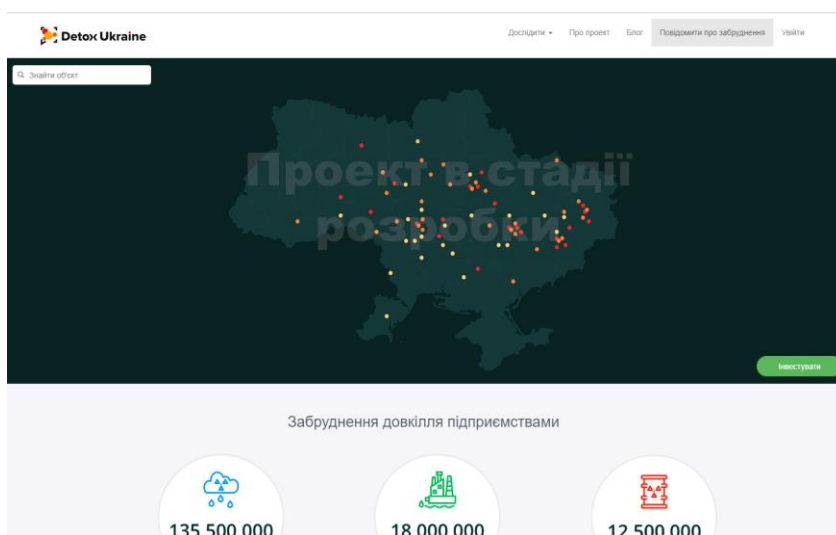


рис.6 Detox Ukraine

### 3) EcoHolmes

EcoHolmes - веб-сервіс для відстеження розподілу та раціональності використання бюджетних коштів від екологічного податку. Проект став фіналістом Open Data Challenge 2019, але інформації про нього зовсім немає.

### 4) EcoViewer

EcoViewer [2] - веб-сервіс, який інформує про стан забруднення повітря промисловими об'єктами та відображає його вплив на здоров'я людини. Більша частина функцій сервісу також не працює.



Рис. 7 EcoViewer

**Метою статті** є аналіз переваг отримання даних з OpenData за допомогою технології GraphQL та порівняння її з REST.

**Виклад основного матеріалу дослідження.** Технології REST і GraphQL, є архітектурами розробки API, які можна використовувати для створення веб-служб для керованих даними додатків.

GraphQL – це мова запитів з відкритим вихідним кодом, що розроблена компанією Facebook. Вона створювалася як більш ефективна альтернатива REST для розробки та використання програмних інтерфейсів. GraphQL не прив'язана до будь-якої конкретної бази даних або механізму зберігання даних.

Facebook виклав у вільний доступ специфікацію GraphQL і реалізацію на мові JavaScript. З тих пір ця специфікація була реалізована на багатьох інших великих мовах програмування. Крім того, екосистема, що склалася навколо GraphQL,

зростає не тільки горизонтально, поширюючись на інші мови програмування, але і вертикально (поверх GraphQL надбудовуються бібліотеки, наприклад, Apollo, Relay).

REST (Representational state transfer) - це API (Application Programming Interface - інтерфейс прикладного програмування), який надає можливість клієнт-серверної комунікації для веб-додатків по протоколу HTTP. API працює за принципом CRUD (POST (Create) – створення, GET (Read) - читання, PUT (Update) - оновлення, DELETE (Delete) – видалення), які відповідають основним операціям в SQL базах даних: INSERT, SELECT, UPDATE, and DELETE. Підхід REST завжди обмежений для роботи з єдиними ресурсами. Якщо потрібні дані, які надходять від двох або більше ресурсів (наприклад, повідомлень і користувачів), то потрібно здійснити кілька запитів на сервер. Крім того, REST-запити завжди повертають повний набір даних, доступних для певного ресурсу. Немає можливості обмежити запит лише для отримання підмножини полів даних.

Приклад алгоритму отримання даних з ресурсу OpenData за допомогою REST, що запропоновано на офіційному сайті:

1. Зайти під паролем користувача та знайти відповідний набір даних.
2. Внизу на сторінці набору даних в таблиці «Додаткової інформації» знайти «Ідентифікаційний номер набору даних», наприклад, e30e47e2-8fd0-4202-a5d4-20f07cfd116c.
3. Після цього зробити запит API за адресою [https://data.gov.ua/api/3/action/package\\_show?id=Ідентифікаційний-код-набору](https://data.gov.ua/api/3/action/package_show?id=Ідентифікаційний-код-набору), тобто [https://data.gov.ua/api/3/action/package\\_show?id=e30e47e2-8fd0-4202-a5d4-20f07cfd116c](https://data.gov.ua/api/3/action/package_show?id=e30e47e2-8fd0-4202-a5d4-20f07cfd116c) . Цей запит поверне всю мета-інформацію про набір даних та його ресурсні файли, його вага 14.39 кБ.
4. Далі, в отриманому JSON знайти масив «resources», останній запис якого буде найновішим ресурсом з набору даних, але інші файли також містять

актуальну інформацію, тож для того, щоб отримати всю інформацію, потрібно буде зробити більше запитів.

5. Взяти значення id ресурсу та робимо запит до функції API [https://data.gov.ua/api/3/action/resource\\_show?id=ID-ресурсу](https://data.gov.ua/api/3/action/resource_show?id=ID-ресурсу),  
[https://data.gov.ua/api/3/action/resource\\_show?id=b0476139-62f2-4ede-9d3b-884ad99afd08](https://data.gov.ua/api/3/action/resource_show?id=b0476139-62f2-4ede-9d3b-884ad99afd08), вага - 5.08 кБ.
6. Отримати JSON масив, який містить інформацію про ресурси. Виділити з масиву поле «url», в якому є посилання на файл з ресурсом, наприклад [https://data.gov.ua/dataset/2cde453d-a726-40e8-95f5-03eb05d4bfcc/resource/c6a2c91c-9e3e-48ba-b5c3-887faa3674fc/revision/23102019\\_1](https://data.gov.ua/dataset/2cde453d-a726-40e8-95f5-03eb05d4bfcc/resource/c6a2c91c-9e3e-48ba-b5c3-887faa3674fc/revision/23102019_1)

Тобто потрібно зробити мінімум 3 запити, щоб отримати інформацію, але разом з цією інформацією прийде багато надлишкових даних, що даремно використовують трафік користувача і збільшує час виконання запитів.

Підхід GraphQL набагато гнучкіший і здатний подолати основні недоліки REST. За допомогою мови запитів GraphQL можна точно описати, як має виглядати відповідь. Розробник може вказати, які поля повинні бути включені, щоб обмежити відповідь на потрібні дані. Крім того, є можливість використовувати графік і об'єднати об'єднані об'єкти в межах одного запиту даних GraphQL. Немає необхідності у додаткових кругових запитах на сервері.

Основними перевагами GraphQL є те, що розробник отримує інформацію саме в тому обсязі, в якому хоче отримати, на відміну від REST, де відповідь на запит може містити багато непотрібних даних. Потрібна всього одна кінцева точка, ніяких додаткових версій для унікального API.

На жаль, OpenData поки не може повноцінно працювати з GraphQL, адже для цього потрібен окремо розроблений бекенд для даної технології, але за допомогою використання бібліотеки `apollo-link-rest` можна працювати з інформацією так, ніби сервер має API, що працюють з технологією GraphQL.





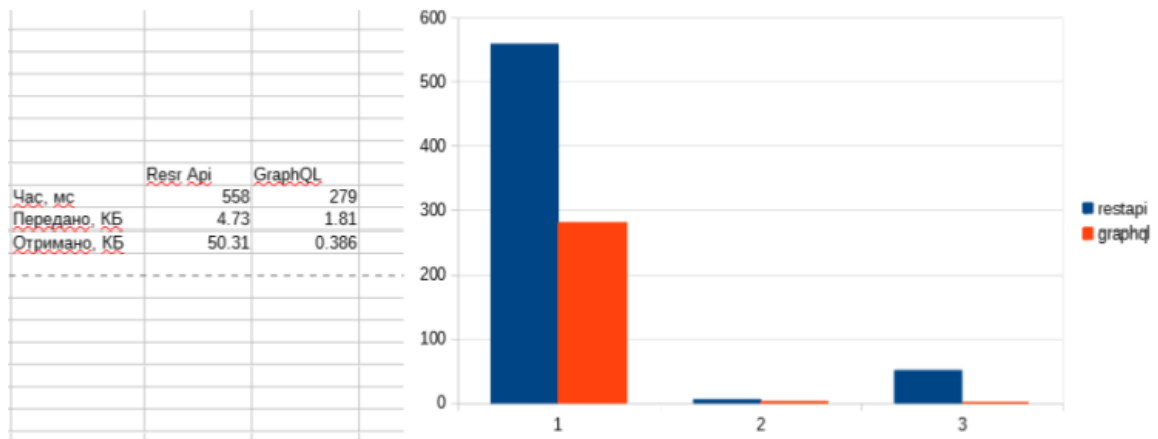


Рис.11 Порівняння результатів виконання запитів REST та GraphQL

Щоб отримати дані за допомогою GraphQL, потрібно повторити перші два кроки з інструкції, що була наведена вище і відправити лише один запит (приклад запиту показано на рис.8), який містить всі актуальні посилання на дані набору.

Перед тим, як відправити запит, потрібно ініціалізувати змінні `restLink` (використовується для того, щоб отримати GraphQL запит від сервера, що працює з REST) та `ApolloClient` (використовується для спрощення роботи з GraphQL). Приклад коду ініціалізації цих змінних з використанням фреймворка `Vue`:

```
const restLink = new RestLink({
  uri: "https://cors-anywhere.herokuapp.com/https://data.gov.ua/",
});
const client = new ApolloClient({
  link: restLink,
  cache: new InMemoryCache(),
});
const apolloProvider = new VueApollo({
  defaultClient: client,
})
```

### **Висновки та перспективи подальших пошуків у напрямі дослідження.**

Внаслідок проведеного аналізу можна зробити висновок, що для додатку, на основі відкритих даних раціональніше використати технологію GraphQL.

Розроблений додаток `OpenEcology` для ознайомлення з радіаційним та екологічним станом в Україні на основі даних з `OpenData` має зручний інтерфейс,

гарний зовнішній вигляд і показує актуальні дані, не потребуючи втручання. В додатку враховано і виправлено основі недоліки схожих сайтів.

В перспективі проект можна оптимізувати, коли OpenData матиме повноцінний бекенд для використання GraphQL. Також проект задовольняє всі критерії проекту Open Data Challenge, тож може бути покращений в рамках цього конкурсу.

### **Список літератури**

1. Detox Ukraine [Електронний ресурс] // detoxukraine – Режим доступу: <http://www.detoxukraine.org/>
2. EcoViewer [Електронний ресурс] // ecoviewer – Режим доступу: <https://www.ecoviewer.info/>
3. GraphQL: A data query language [online]. Lee Byron, 2015 – Режим доступу: <https://code.facebook.com/posts/1691455094417024/graphql-a-data-query-language/>.
4. Зелений Кропивницький [Електронний ресурс] // texty – Режим доступу: <http://texty.org.ua/d/2018/trees/>