

Міністерство освіти і науки України

Національний університет водного господарства
та природокористування

Навчально-науковий інститут агроекології та землеустрою

Кафедра геодезії та картографії

05-04-160М

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних і самостійних робіт з навчальної
дисципліни **«Проектування та управління базами
геопросторових даних» (частина 3)** для здобувачів вищої
освіти першого (бакалаврського) рівня за освітньо-професійною
програмою «Геодезія та землеустрій» спеціальності
193 «Геодезія та землеустрій» усіх форм навчання

Рекомендовано науково-методичною
радою з якості ННІАЗ
Протокол №13 від 18 лютого 2025 р.

Рівне – 2025

Методичні вказівки до виконання лабораторних і самостійних робіт з навчальної дисципліни «Проектування та управління базами геопросторових даних» (частина 3) для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Геодезія та землеустрій» спеціальності 193 «Геодезія та землеустрій» усіх форм навчання [Електронне видання] / Янчук О. Є., Прокопчук А. В. – Рівне : НУВГП, 2025. – 38 с.

Укладачі:

Янчук О. Є., канд. техн. наук, доцент кафедри геодезії та картографії;

Прокопчук А. В., ст.викл. кафедри геодезії та картографії.

Відповідальний за випуск:

Янчук Р. М., к.т.н., доцент, завідувач кафедри геодезії та картографії.

Керівник групи забезпечення спеціальності:

Янчук Р. М., к.т.н., доцент, завідувач кафедри геодезії та картографії.

© О. Є. Янчук,

А. В. Прокопчук, 2025

© Національний університет
водного господарства та
природокористування, 2025

ЗМІСТ

ПЕРЕДМОВА	4
ЛАБОРАТОРНІ РОБОТИ.....	5
Лабораторна робота № 8 Робота з Geoserver	5
Лабораторна робота № 9 Компонування WMS карти з використанням java-бібліотеки Leaflet	20
Лабораторна робота № 10 Компонування WFS карти з використанням java-бібліотеки Leaflet	27
РЕКОМЕНДОВАНА ЛІТЕРАТУРА ТА РЕСУРСИ.....	38

ПЕРЕДМОВА

Методичні вказівки складено відповідно до програми навчальної дисципліни «Проектування та управління базами геопросторових даних» та призначено для здобувачів вищої освіти першого (бакалаврського) рівня спеціальності 193 «Геодезія та землеустрій» усіх форм навчання.

Головним завданням курсу є навчання студентів навичкам проектування, організації функціонування і використання баз даних геоінформаційних систем, а також впровадження таких систем в виробництво і органи державного управління та у всі можливі галузі народного господарства.

У даній частині методичних вказівок розглянуто принципи веб-публікації геопросторової інформації з бази даних PostgreSQL. Описано послідовність публікації даних на сервері Geoserver та компонування веб-карти з використанням бібліотеки Leaflet.

У результаті виконання представлених лабораторних робіт студенти повинні навчитися публікувати просторову інформацію з бази даних PostgreSQL/Postgis у середовищі Geoserver, групувати шари та налаштовувати вигляд опублікованих даних, компонувати html-сторінки з веб-картами, налаштовувати вигляд об'єктів на таких геопорталах, легенду, додаткові інструменти.

ЛАБОРАТОРНІ РОБОТИ

Лабораторна робота № 8

Робота з Geoserver

Мета: навчитись публікувати дані у середовищі Geoserver.

Завдання: опублікувати всі шари з бази даних Postgis у Geoserver; згрупувати їх на перегляд разом; налаштувати стилі відповідно до оформлення у QGIS; опублікувати растрову підкладку (при її відсутності – спочатку завантажити фрагмент через SAS Planet); підключитися до опублікованих на Geoserver шарів з QGIS через WFS-сервіс.

1) Публікація векторних даних з Postgis у Geoserver

Geoserver – це веб-сервер із відкритим кодом, який дозволяє користувачам обмінюватися геопросторовими даними. Відповідає стандартам Open Geospatial Consortium: WMS, WFS, WCS, WPS (рис. 1.1). Забезпечує стандартним клієнтам, таким, як веб-браузери та настільні ГІС, доступ до карт і даних, що зберігаються у різних форматах. Це означає, що ваші просторові дані можуть зберігатися практично в будь-якому зручному вам форматі. У найпростішому випадку для користувачів ваших даних достатньо мати веб-браузер, щоб переглянути дані.

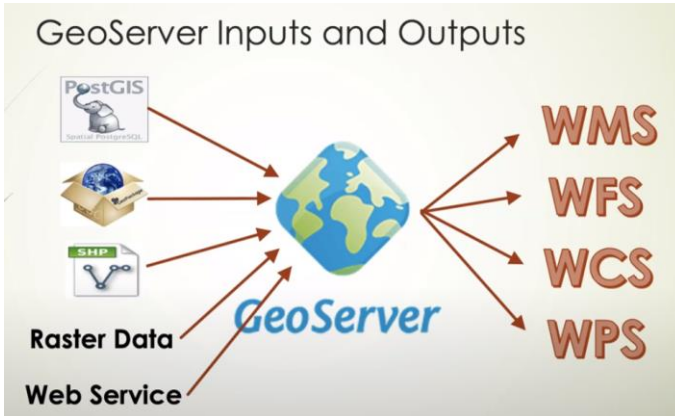


Рис. 1.1. Вхідні та вихідні дані у Geoserver

Стандарт *WFS (Web Feature Service)* визначає інтерфейси та операції, які дозволяють запитувати та редагувати векторні просторові дані. Сервіс *WFS* приховує справжні сховища даних і програма-клієнт отримує потік векторних даних, при цьому не знаючи взяті вони з БД або з файлу. Основне завдання *WFS*-сервісів – це надання користувачам універсального інтерфейсу доступу до просторових даних, що прибирає необхідність прямого доступу до сховища.

Стандарт *WCS (Web Coverage Service)* визначає інтерфейси і операції, що дозволяють взаємодіяти з просторовими даними, які описуються термінами грід (grid) або покриття (coverage). Цим терміном описуються супутникові знімки, результати аерофотозйомки, дані з багатоканальних сенсорів ДЗЗ, моделі рельєфу (DEM) та інші дані. Всі вони мають схожий формат зберігання – область, представлена регулярною сіткою, в кожному осередку якої міститься кілька значень.

Стандарт *WMS (Web Map Service)* забезпечує інтерфейс для отримання геоприв'язаних зображень карт з однієї або кількох розподілених геопросторових баз даних. Запит *WMS* визначає географічний шар(и) і територію інтересу для обробки. Відповіддю на запит є одне або кілька географічно зареєстрованих зображень карти (у форматі JPEG, PNG тощо), які можна відобразити у браузері або настільному додатку.

Стандарт *WPS (Web Processing Service)* містить правила для стандартизації вхідних і вихідних даних (запитів і відповідей) для послуг геопросторової обробки

Geoserver є вільним програмним забезпеченням і встановлюється з офіційного сайту <https://geoserver.org/download/>

Послідовність публікації векторних даних у Geoserver

1. Запускаємо браузер та у адресному рядку вводим адресу <http://localhost:8080/geoserver/web/>

У даному випадку *localhost* – означає, що сервер на якому розташований Geoserver, знаходиться на тому ж комп'ютері з якого виконується підключення. При необхідності підключитися

до іншого комп'ютера необхідно ввести його ір-адресу. 8080 – стандартний порт, через який по замовчуванню здійснюється доступ до Geoserver.

2. У верхній частині вікна необхідно **ввести логін та пароль**, задані при встановленні програми.

За замовчуванням:

Логін: *admin*

Пароль: *geoserver*

3. Додаємо новий робочий набір (Workspace). У лівій частині вікна оберіть **Workspaces / Add new workspace**.

4. Задайте **ім'я** та **url адресу** для звернення до робочого набору та натисніть **Submit** (рис. 1.2).

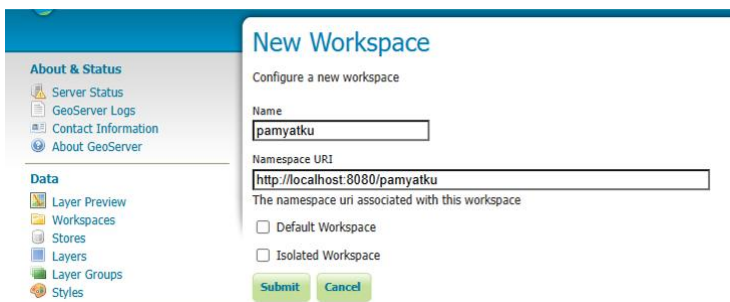


Рис. 1.2. Опис робочого набору

5. Додаємо нове сховище даних (Stores). У лівій частині вікна оберіть **Stores / Add new Store / Postgis** (рис. 1.3).

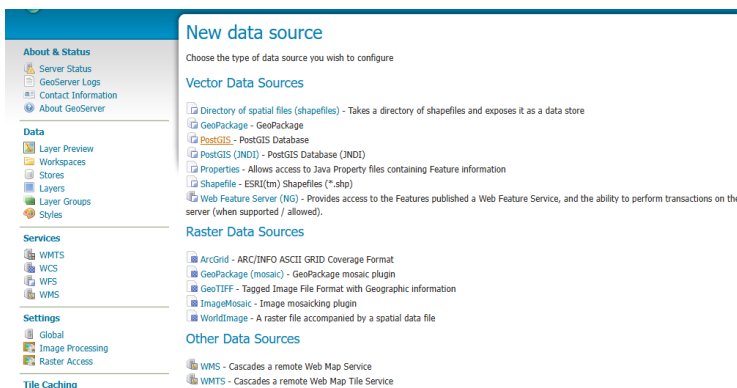


Рис. 1.3. Підключення сховища даних Postgis

6. Налаштовуємо підключення до сховища (у даному випадку бази геоданих): обираємо потрібний **робочий простір**; задаємо **назву джерела даних** (довільна); задаємо параметри з'єднання з БД Postgis – **ІР, порт, назва БД, користувач, пароль** (необхідно вводити логін та пароль саме користувача Postgis, а не Geoserver) (рис. 1.4). Решту налаштувань залишаємо за замовчуванням та натискаємо **Save**.

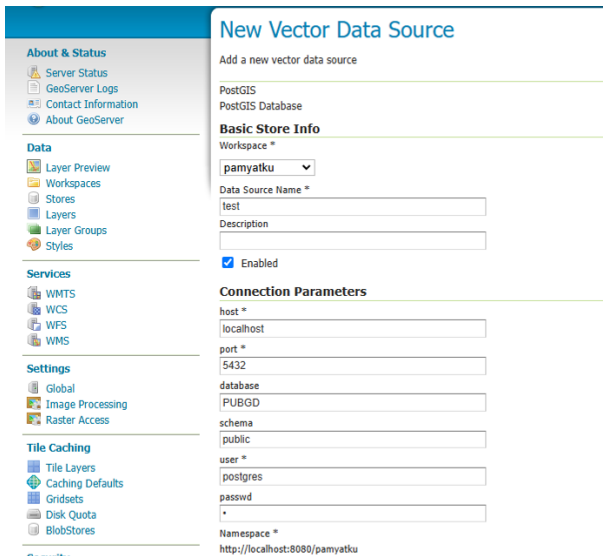


Рис. 1.4. Налаштування підключення до сховища даних Postgis

7. У наступному вікні відображається перелік шарів, які містяться у базі геоданих (рис. 1.5). Для публікації шару на Geoserver натискаємо навпроти нього **Publish**.

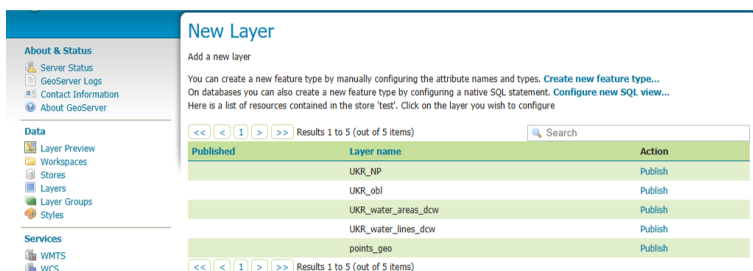


Рис. 1.5. Перелік шарів доступних для публікації

8. Налаштовуємо параметри публікації обраного шару: у блоці **Coordinate Reference Systems** задаємо/перевіряємо кінцеву систему координат (**Declared SRS**); у блоці **Lat/Lon Bounding Box** натискаємо кнопку **Compute from native bounds**, щоб обчислити координати обмежуючого прямокутника для району робіт (рис. 1.6). Решту налаштувань залишаємо за замовчуванням та натискаємо **Save**.

Coordinate Reference Systems

Native SRS
 EPSG:4326 EPSG:WGS 84...

Declared SRS
 EPSG:4326 Find... EPSG:WGS 84...

SRS handling
 Force declared ▼

Bounding Boxes

Native Bounding Box

Min X	Min Y	Max X	Max Y
26.498692752824	44.612550499829	35.074776178293	50.452949105851

[Compute from data](#)
[Compute from SRS bounds](#)

Lat/Lon Bounding Box

Min X	Min Y	Max X	Max Y
26.498692752824	44.612550499829	35.074776178293	50.452949105851

[Compute from native bounds](#)

Рис. 1.6. Налаштування шару для публікації

9. Шар опубліковано на Geoserver. Для перегляду результату ліворуч обираємо **Layer Preview** та навпроти назви опублікованого шару натискаємо **OpenLayers**. Клацнувши на об'єкті можна переглянути його атрибути внизу карти (рис. 1.7).

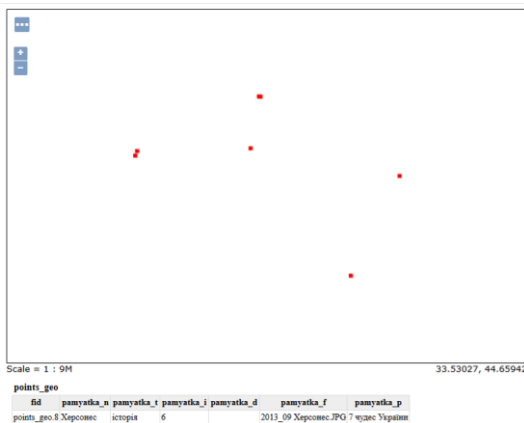


Рис. 1.7. Перегляд опублікованого шару

Клацнувши на “трикрапку” у верхньому лівому куті можна відкрити параметри карти та змінити їх.

10. Для публікації інших шарів з бази Postgis обираємо **Layers / Add a new layer** та з випадаючого списку обираємо підключення створене у пункті 6 (рис. 1.8). Натискаємо **Publish** навпроти шару який хочемо опублікувати та задаємо налаштування, як вказано у пункті 8.

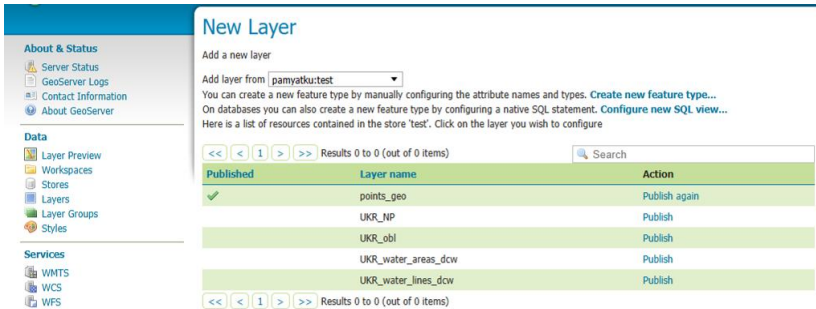


Рис. 1.8. Перелік шарів опублікованих та доступних для публікації

Таким чином публікуємо на Geoserver всі шари, які містяться у базі геоданих.

2) Групування шарів у Geoserver

Групування – це процес об’єднання шарів геоданих з метою їх комплексної візуалізації.

Для групування:

1. Обираємо ліворуч **Layer Groups / Add new layer group**.
2. Задаємо довільне **ім'я групи, заголовок** та з випадаючого списку обираємо з яким **робочим простором** працюємо (рис. 1.9).

3. У блоці **Layers** натискаємо **Add Layer...** та вибираємо шари для додавання у групу. Стрілочками редагуємо **порядок відображення шарів** (*шар розміщений у переліку нижче публікується пізніше, і буде відображений поверх попередніх*) (рис. 1.9).

4. У блоці **Bounds** генеруємо межі фрейму з вмісту шарів натиснувши **Generate Bounds** (рис. 1.9). Решту налаштувань залишаємо за замовчуванням та натискаємо **Save**.

Name

Enabled
 Advertised

Title i18n

Abstract i18n

Workspace

Bounds
 Min X Min Y Max X Max Y

Coordinate Reference System
 EPSG:WGS 84...

Mode

Queryable

Layers

Drawing order	Type	Layer	Default Style	Style	Remove
1 <input type="button" value="⌵"/>	Layer	pamyatku:UKR_obl	<input type="checkbox"/>	polygon	<input type="button" value="⊖"/>
2 <input type="button" value="⬆️"/> <input type="button" value="⬇️"/>	Layer	pamyatku:UKR_NP	<input type="checkbox"/>	polygon	<input type="button" value="⊖"/>
3 <input type="button" value="⬆️"/> <input type="button" value="⬇️"/>	Layer	pamyatku:UKR_water_areas_dcw	<input type="checkbox"/>	polygon	<input type="button" value="⊖"/>
4 <input type="button" value="⬆️"/> <input type="button" value="⬇️"/>	Layer	pamyatku:UKR_water_lines_dcw	<input type="checkbox"/>	line	<input type="button" value="⊖"/>
5 <input type="button" value="⬆️"/>	Layer	pamyatku:points_geo	<input type="checkbox"/>	point	<input type="button" value="⊖"/>

Рис. 1.9. Налаштування групи шарів

Для перегляду групи шарів обираємо **Layer Preview** та навпроти назви групи натискаємо **OpenLayers**. Клацнувши на об'єкті можна переглянути його атрибути внизу карти (рис. 1.10).

WMS version: 1.1.1 Tiling: Single tile Antialias: Full Format: PNG 24bit Styles: Default Width/Height: 950 400

Filter: CQL

Scale = 1 : 9M

fid	type	name_ua	name_lat	koatuu	codetopo	codename	label	shape_leng	shape_area
UKR_obl.22	Області, автономії та ради	Хмельницька	Khmelnytska	6800000000	81210000	Межі областей та автономії	Хмельницька область	12.11214826	2.56042282631

fid	name_ua_se	name_lat_s	koatuu_set	codename_s	population	name_ua_b1	name_lat_1	koatuu_b_1	name_ua_1	nam
UKR_NP.21189	Дубина	Dubyna	6824288902	село	63.0	Старокостянтинівський	Starokostiantynivskyi	6824200000	Хмельницька	Khme
UKR_NP.21385	Грибенювка	Hrybenyuka	6824286402	село	338.0	Старокостянтинівський	Starokostiantynivskyi	6824200000	Хмельницька	Khme
UKR_NP.21402	Жабче	Zhabche	6824283209	село	339.0	Старокостянтинівський	Starokostiantynivskyi	6824200000	Хмельницька	Khme

Рис. 1.10. Перегляд опублікованої групи шарів

3) Налаштування стилів оформлення у Geoserver через QGIS

Для налаштування вигляду об'єктів шару використовуються стилі. Найпростіше створити стиль оформлення в QGIS та зберегти його у форматі *.sld.

Стандарт *SLD (Styled Layer Descriptor)* призначений для опису символіки просторових даних. Зазвичай застосовується в сервісах Web Map Service (WMS) або Web Feature Service (WFS).

Для налаштування стилю у QGIS відкриваємо **необхідний шар**, заходимо у його **Властивості** на вкладку **Символіка** та задаємо бажане оформлення шару.

Далі у нижній частині вікна обираємо **Стиль / Зберегти стиль**, з випадаючого списку обираємо **As SLD Style File** та задаємо місце збереження (рис. 1.11).

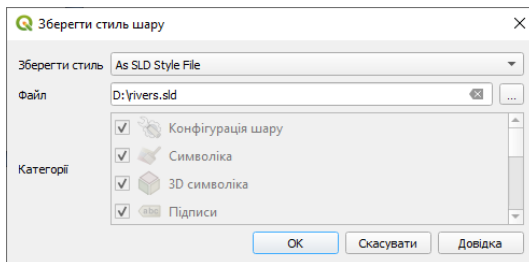
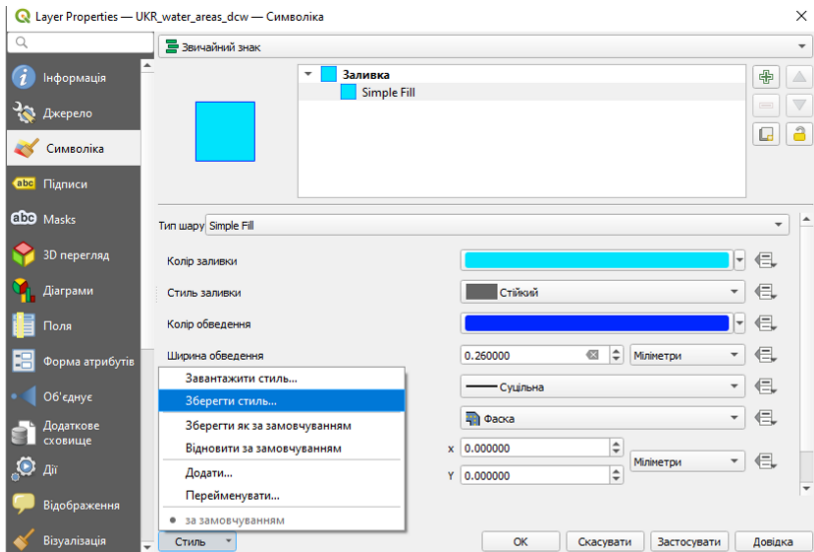


Рис. 1.11. Збереження стилю оформлення

Для публікації створеного стилю в Geoserver обираємо **Styles / Add a new style**. Задаємо **ім'я стилю**, з випадючих списків обираємо **робочий набір** та **формат стилю**. Далі у блоці **Upload a style file** обираємо файл з описом стилю, натискаємо **Upload...** У нижній частині вікна **Validate** та у разі відсутності помилок – **Save** (рис. 1.12).

Після публікації стилю необхідно його підключити для конкретного шару. Обираємо **Layers**, клацаємо на **потрібному шарі**, переходимо на вкладку **Publishing** та у блоці **WMS Settings / Layer Settings** обираємо з випадючого списку **Default Style** потрібний стиль. Для збереження змін натискаємо **Save**.

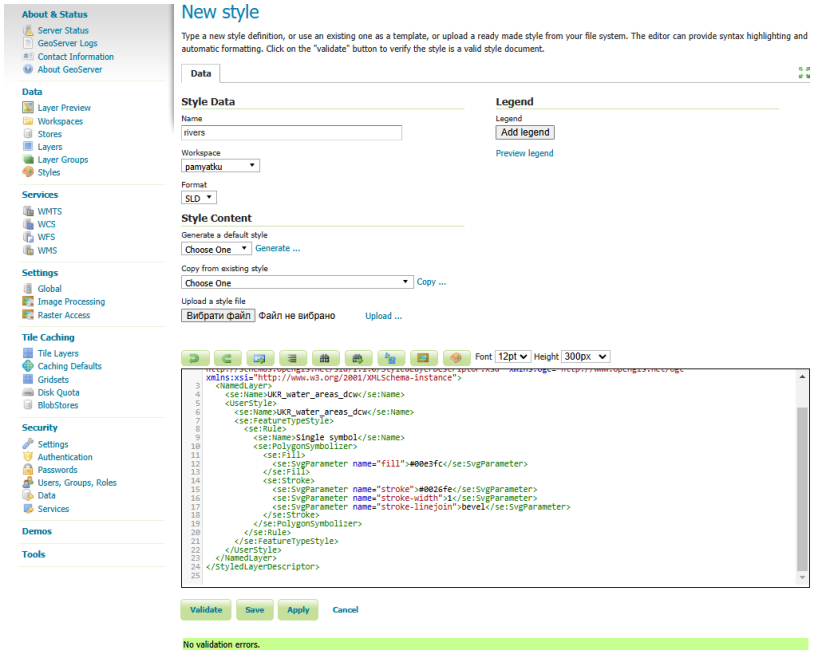


Рис. 1.12. Публікація стилю в Geoserver

Результат можна переглянути перейшовши у **Layer Preview** та натиснувши **OpenLayers** навпроти потрібного шару (рис. 1.13).

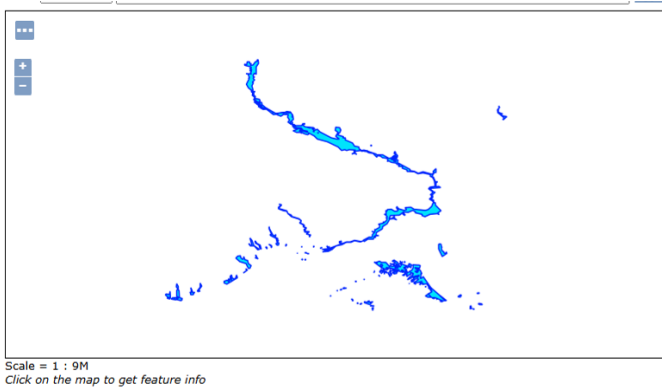


Рис. 1.13. Перегляд шару з підключеним стилем оформлення

Опубліковані стилі можна використовувати і для оформлення шарів у групах. Обираємо **Layer Groups**, клацаємо на **потрібній групі**, та у блоці **Layers** у колонці **Style** обираємо **потрібний стиль оформлення** для відповідного шару (рис. 1.14). Для збереження змін натискаємо **Save**.

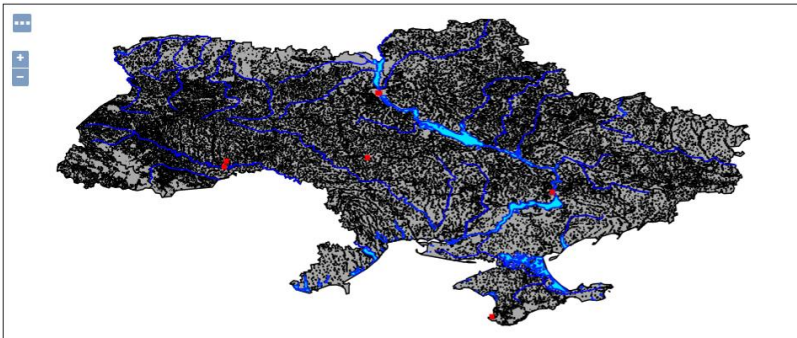
Layers

[Add Layer...](#)
[Add Layer Group...](#)
[Add Style Group...](#)

Drawing order	Type	Layer	Default Style	Style	Remove
1	Layer	пам'ятку:UKR_obl	<input type="checkbox"/>	polygon	
2	Layer	пам'ятку:UKR_NP	<input type="checkbox"/>	polygon	
3	Layer	пам'ятку:UKR_water_areas_dcw	<input type="checkbox"/>	rivers	
4	Layer	пам'ятку:UKR_water_lines_dcw	<input type="checkbox"/>	line	
5	Layer	пам'ятку:points_geo	<input type="checkbox"/>	point	

Рис. 1.14. Підключення стилю у групі для шару полігональних водних об'єктів

Результат можна переглянути перейшовши у **Layer Preview** та натиснувши **OpenLayers** навпроти потрібної групи шарів (рис. 1.15).



Scale = 1 : 9M
Click on the map to get feature info

Рис. 1.15. Перегляд групи шарів з підключеним стилем оформлення для шару полігональних водних об'єктів

4) Публікація растрових даних у Geoserver

Для підключення растрових даних додаємо нове сховище даних (Stores). У лівій частині вікна оберіть **Stores / Add new Store / Raster Data Sources / GeoTIFF - Tagged Image File Format with Geographic information** (рис. 1.3).

Налаштуємо підключення до сховища (у даному випадку до растрового файлу на локальному комп'ютері): обираємо потрібний **робочий простір**, задаємо **назву джерела даних** (довільна) та у блоці **Connection Parameters** вказуємо **розташування** растрового файлу (рис. 1.16). Натискаємо **Save**.

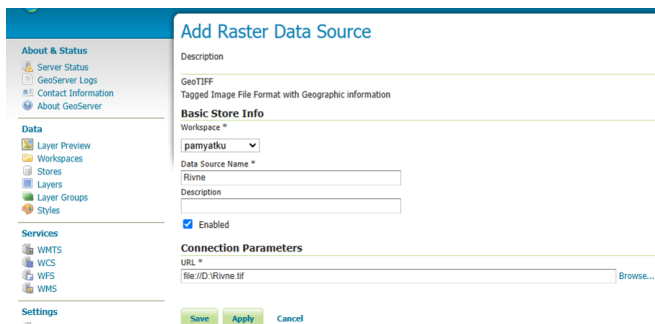


Рис. 1.16. Підключення растрового джерела даних

У наступному вікні відображається назва растрового шару, який можна опублікувати натиснувши **Publish**.

З'являється вікно з налаштуванням параметрів публікації. Перевіряємо назву шару, початкову та цільову системи координат, координати обмежуючого прямокутника та натискаємо **Save**.

Переглянути опублікований растр можна перейшовши на вкладку **Layer Preview** та натиснувши **OpenLayers** навпроти назви опублікованого шару (рис. 1.17).



Рис. 1.17. Перегляд опублікованого растру

Завантаження геоприв'язаного растру через SAS Planet

1. Запустіть програму **SASPlanet**.
2. У головному меню оберіть **Карти / Google / Карта (Google)** та знайдіть на екрані необхідну територію, наприклад м.Рівне.
3. Змініть базову карту на супутникове зображення за допомогою команди **Карти / Google / Супутник (Google)**.
4. Активуйте команду з головної панелі інструментів **Виділити прямокутну область** (або комбінація клавіш **Alt+R**) та **намалюйте прямокутник** навколо м.Рівне. Якщо автоматично не з'явилася вікно з налаштуваннями завантаження, то натисніть комбінацію клавіш **Ctrl+B**.
5. У вікні **Операції з виділеною областю** на вкладці **Завантажити** задайте потрібний **Масштаб**, на вкладці **Склеїти** у рядку **Результуючий формат** задайте **GeoTIFF**, у рядку **Куди зберігати** – бажану адресу збереження файлу, у випадяючому списку **Масштаб** ще раз продублюйте обраний масштаб та у блоці **Створити файл прив'язки** відмітьте позицію **w** (рис. 1.18). Натисніть **Почати**.

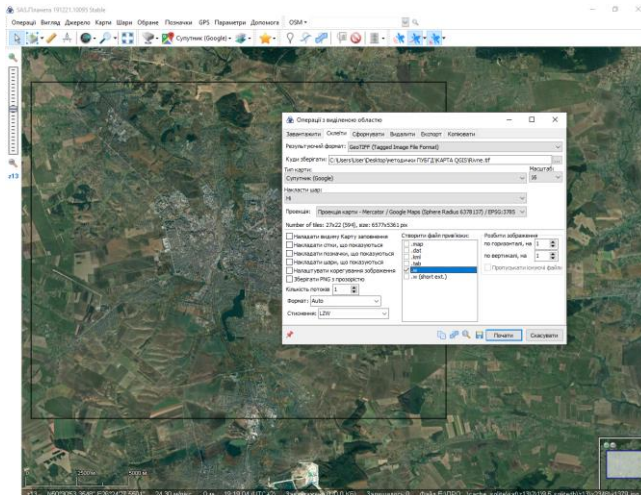


Рис. 1.18. Налаштування завантаження фрагменту супутникового растрового зображення

5) Доступ до опублікованих даних через WFS-сервіс

Шари опубліковані на Geoserver можна підключати у QGIS як WFS-сервіс.

Переваги такого підключення:

- доступність атрибутивних даних;
- можливість редагування векторних та атрибутивних даних;
- можливість роботи у багатокористувацькому режимі;
- робота з об'єктами БД без прямого підключення до БД, що зменшує ймовірність пошкодження самої бази даних, адже немає потреби надавати прямий доступ до портів комп'ютера.

Для підключення робочого набору з Geoserver у QGIS як WFS-сервісу:

1. У браузері QGIS шукаємо пункт **WFS/OGC API – Features**.

2. З контекстного меню обираємо **Створити з'єднання**.

3. Задаємо довільне ім'я з'єднання та вводимо адресу WFS-сервісу скопійовану з Geoserver, зі сторінки перегляду вмісту шару (замінивши у адресі wms на wfs), наприклад <http://localhost:8080/geoserver/pamyatku/wfs> (рис. 1.19).

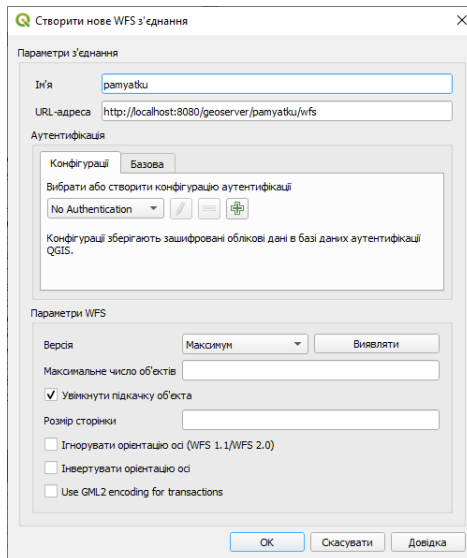


Рис. 1.19. Налаштування підключення WFS-сервісу

У браузері об'єктів обираємо необхідні шари для відображення у програмі (рис. 1.20).

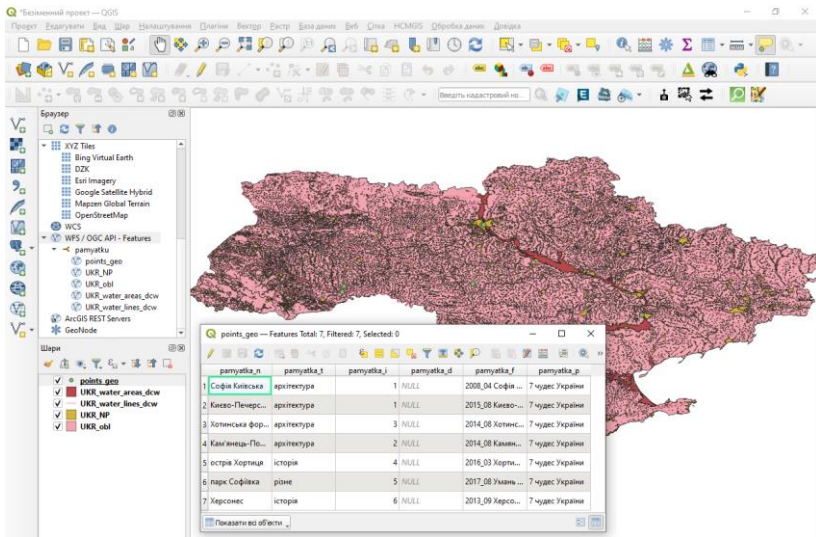


Рис. 1.20. Перегляд об'єктів БД через WFS підключення

Завдання для самостійної роботи. Розглянути налаштування стилів у Geoserver через Geoserver CSS extension.

Запитання для контролю:

1. Як встановити Geoserver?
2. Як запустити Geoserver?
3. Як підключити до Geoserver сховище з бази даних Postgis?
4. Як опублікувати на Geoserver шар з бази даних Postgis?
5. Як налаштувати та підключити стиль оформлення шару у Geoserver?
6. Як переглянути опублікований шар на Geoserver?
7. Як згрупувати шари на Geoserver?
8. Як опублікувати на Geoserver растровий шар?
9. Як завантажити растр через SASPlanet?
10. Як підключити шар у QGIS як WFS-сервіс?

Лабораторна робота № 9

Компонування WMS карти з використанням java-бібліотеки Leaflet

Мета: навчитись компонувати веб-сторінку на основі WMS карти з використанням java-бібліотеки Leaflet.

Завдання: створити веб-сторінку з картою, до якої як WMS-сервіс підключено шари з Geoserver, базові карти, налаштовано легенду та можливість перегляду атрибутів об'єктів.

1) Загальні положення

Веб-сторінка html має чітко визначену структуру. Обов'язковими є чотири парні теги, які записуються у строго визначеній послідовності:

```
<html>
  <head>
    <title>Назва сторінки</title>
  </head>
  <body>
    Вміст сторінки
  </body>
</html>
```

Тег **<head>** призначений для зберігання службової інформації про сторінку, мета якої - допомогти браузеру в роботі з даними (наприклад кодування сторінки, таблиця стилів). Тег **<title>** містить заголовок сторінки, який відображається на закладці браузера. Безпосередньо вміст html-документу міститься між тегами **<body>...</body>**.

Для компонування карт зручно використовувати існуючі бібліотеки типу Leaflet, GeoExt. Для їх використання – на початку тегу **<body>** слід прописати підключення потрібних java-скриптів бібліотек. Після цього розміщується скрипт із вмістом створюваної карти. Таким чином ми компонуємо з необхідних “блоків” свою карту налаштовуючи її вигляд та вміст. В якості вихідних даних до цієї й наступної лабораторних

робіт використовується шаблон html-сторінки з підключеними бібліотеками, який можна завантажити на сторінці дисципліни у системі Moodle.

Leaflet — це бібліотека з відкритим кодом на основі JavaScript для створення інтерактивних карт. Документацію про команди бібліотеки Leaflet можна знайти на офіційному сайті <https://leafletjs.com/index.html>.

Розглянемо приклади скриптів із застосуванням типових команд бібліотеки Leaflet для підключення базових растрових карт, відображення шарів з бази даних (опублікованих на Geoserver), керування видимістю шарів у легенді тощо.

Варто звернути увагу на декілька загальних принципів:

- команди бібліотеки Leaflet починаються з «**L**.»
- для додавання рядка з коментарем у html-сторінку використовується конструкція **<!--текст коментаря-->**
- для додавання рядка з коментарем у скрипт використовується конструкція з двох нахилених рисок **//текст коментаря**
- для додавання декількох рядків з коментарем у скрипт використовується конструкція **/*текст коментаря
текст коментаря */**
- кількість пробілів на початку рядка не впливає на виконання коду.

2) Підключення базових карт

На початку тегу `<body>` слід прописати блок з оголошенням контейнера для розміщення карти, а також задати її ідентифікатор, ширину та висоту.

```
<div id="mapid" style="width: 1200px; height: 800px;">  
</div>
```

Таким чином тег `<body>` містить оголошення контейнера з картою, підключення java-скриптів бібліотек та власне скрипт `<script>`, де описується вміст карти (рис. 2.1).

```

9 <body> <!--Початок тіла сторінки. Те, що бачить користувач.-->
10 <div id="mapid" style="width: 1200px; height: 800px;"></div> <!--Створення контейнер
висоту-->
11
12 <!--Підключення JAVA-скриптів бібліотек: Leaflet, jquery, BetterWMS, Leaflet (AJAX)
13 <script src="leaflet/leaflet.js"></script>
14 <script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
15 <script src="L.TileLayer.BetterWMS.js" charset="utf-8"></script>
16 <script type="text/javascript" src="dist/leaflet.ajax.js"></script>
17
18 <script> <!--Початок вашого скрипта
19 var мумар = L.map('mapid').setView([50.6, 26.2], 12); <!--Створення карти, котру
Також в квадратних дужках задано координати центру карти, далі вказано базовий м
20 var osm = L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
21     maxZoom: 18,
22     id: 'mapbox.streets'
23 }).addTo(мумар); <!--Додавання базової карти OpenStreetMap. В круглих дужках 'add

```

Рис. 2.1. Фрагмент коду з вмістом тегу <body>

У тезі <script> за допомогою оператора **var** оголошується змінна карти та задаються її налаштування.

```
var мумар = L.map('mapid').setView([50.6, 26.2], 12);
```

Наведений вище фрагмент коду створює карту, котру буде розміщено в межах контейнера з ідентифікатором 'mapid'. Також в квадратних дужках задано координати центру карти при відкритті, далі вказано базовий масштаб.

Для підключення базової карти необхідно оголосити нову змінну з ім'ям базової карти та командою **L.tileLayer** задати посилання на карту. В кінці командою **addTo** потрібно вказати назву карти, до якої буде приєднуватися даний об'єкт.

```
var osm =
L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png',
{}).addTo(мумар);
```

При підключенні базової карти можна використовувати додаткові параметри, наприклад, масштаб в межах якого допускається перегляд підключеної карти, підписи тощо.

```
var osm =
L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png',
{ minZoom: 5, maxZoom: 22,
attribution: 'Map data © <a
href="http://openstreetmap.org">OpenStreetMap</a>
contributors'
}).addTo(мумар);
```

У рамках лабораторної роботи також можна використати посилання на базові карти (<https://www.geohowtos.com/apps/xyz-tiles/qgis.html>):

OpenTopoMap 'https://{s}.tile.opentopomap.org/{z}/{x}/{y}.png'

Google hybrid

'http://mt1.google.com/vt/lyrs=y&x={x}&y={y}&z={z}'

Google standard roadmap

'http://mt1.google.com/vt/lyrs=m&x={x}&y={y}&z={z}'

Esri WorldImagery

'https://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}'

Можна підключати одночасно декілька різних базових карт. Керування їх видимістю здійснюється за допомогою легенди карти, як буде описано далі.

3) Підключення шарів з Geoserver через WMS-сервіс

Шари даних опубліковані на Geoserver можна підключати до створюваної карти через WMS-сервіс. Це дає можливість бачити «картинку» з даними, без перегляду їх атрибутів.

Спочатку необхідно прописати URL-адресу робочого набору Geoserver, де опубліковані дані. Її можна скопіювати з Geoserver, зі сторінки перегляду вмісту шару.

```
var url_wms =  
'http://localhost:8080/geoserver/pamyatku/wms';
```

Після цього прописуємо параметри підключення до потрібних шарів. Оголошуємо змінну з ім'ям шару та командою **L.tileLayer.wms** підключаємо доступ до робочого набору з даними. Обов'язково необхідно зазначити назву шару layers (робочий набір та шар, так як вони називаються на Geoserver), формат, прозорість та вказати назву карти, до якої буде приєднуватися даний об'єкт.

```
var meza = L.tileLayer.wms(url_wms, {  
    layers: 'pamyatku:UKR_obl',  
    format: 'image/png',  
    transparent: true  
}).addTo(мумап);
```

Можна підключати одночасно декілька різних шарів з даними. На карті вони відобразатимуться у порядку згадування у легенді.

З використанням, для підключення шару, команди **L.tileLayer.betterWms** є можливість **переглядати атрибути об'єктів у спливаючому вікні**. Відображаються лише атрибути об'єкта, що відповідає координатам пікселя на якому клацнули курсором.

```
var points = L.tileLayer.wms(url_wms, {  
    layers: 'пам'ятку:points_geo',  
    format: 'image/png',  
    transparent: true  
}).addTo(myMap);
```

Налаштування спливаючого вікна задаються у файлі **L.TileLayer.BetterWMS**. За вміст спливаючого вікна відповідає наведена нижче частина скрипту. В останньому рядку наведеного фрагмента зазначається заголовок спливаючого вікна, бажані назви атрибутів та у квадратних дужках зазначаються номери стовпців з атрибутами (як вони відображаються у Geoserver) (рис. 2.2).

```
var data1 = data.replace(/<\/td>/g, "");  
data1 = data1.split("<td>");  
data = '<h1>Пам'ятка</h1><br>'+ 'Назва: ' +  
data1[2] + '<br>Тун: ' + data1[3];
```



Рис. 2.2. Вигляд спливаючого вікна при підключенні BetterWMS

Якщо підключення зі спливаючими вікнами відбувається вперше, то необхідно «розкоментити» частину файлу `C:\Program Files\GeoServer\webapps\geoserver\WEB-INF\web.xml` та перезапустити GeoServer. Активувати слід частини файлу наведені на рисунку 2.3.

```
138 <filter>
139 <filter-name>cross-origin</filter-name>
140 <filter-class>org.eclipse.jetty.servlets.CrossOriginFilter</filter-class>
141 <init-param>
142 <param-name>chainPreflight</param-name>
143 <param-value>>false</param-value>
144 </init-param>
145 <init-param>
146 <param-name>allowedOrigins</param-name>
147 <param-value>*</param-value>
148 </init-param>
149 <init-param>
150 <param-name>allowedMethods</param-name>
151 <param-value>GET,POST,PUT,DELETE,HEAD,OPTIONS</param-value>
152 </init-param>
153 <init-param>
154 <param-name>allowedHeaders</param-name>
155 <param-value>*</param-value>
156 </init-param>
157 </filter>

190 <filter-mapping>
191 <filter-name>cross-origin</filter-name>
192 <url-pattern>*</url-pattern>
193 </filter-mapping>
```

Рис. 2.3. Частини файлу `web.xml`, які необхідно активувати

4) Налаштування легенди

Після підключення базових карт та WMS-шарів залишається додати можливість вмикати/вимикати видимість потрібних шарів. Для цього групуємо дані та додаємо легенду. Створюємо дві групи шарів `baseLayers` та `overlays` у яких перелічуємо бажані підписи назв шарів у легенді та назви змінних, якими вони підключаються. Команда **L.control.layers** створює легенду зі сформованими групами. Параметр `collapsed:false` означає, що вікно легенди згортатись не буде (при потребі, щоб вікно легенди згорталось використовується значення параметру `true`).

```
var baseLayers = { "OpenStreetMap": osm,
                  "Google": google };
var overlays = { "Межі областей": meza, "Озера": lakes,
                "Річку": rivers, "Пам'ятки": points };
```

```
L.control.layers(baseLayers, overlays, {collapsed:false}).  
addTo(myMap);
```

Таким чином компонується веб-карта, вигляд якої наведено на рисунку 2.4.



Рис. 2.4. Вигляд скомпонуваної веб-карти

Завдання для самостійної роботи. Розглянути можливості бібліотеки GeoExt для компонування WMS карти.

Запитання для контролю:

1. Яка структура html-сторінки?
2. Що таке Leaflet?
3. Як вписати коментар у код html-сторінки?
4. Де задаються розміри контейнера, що містить карту?
5. Як змінити початкові координати на яких буде відкриватися карта?
6. Як підключити шар з Geoserver як WMS-сервіс?
7. Як налаштувати вміст спливаючого вікна на WMS-карті?
8. Як підключити легенду до веб-карти?

Лабораторна робота № 10

Компонування WFS карти з використанням java-бібліотеки Leaflet

Мета: навчитись компонувати веб-сторінку на основі WFS карти з використанням java-бібліотеки Leaflet.

Завдання: створити веб-сторінку з картою, до якої як WFS-сервіс підключено шари з Geoserver, базові карти, налаштовано легенду та можливість перегляду атрибутів об'єктів, підключено додаткові панелі інструментів.

Для компонування карт використовується шаблон html-сторінки з підключеними бібліотеками, який можна завантажити на сторінці дисципліни у системі Moodle.

Підключення базових карт здійснюються так як описано у питанні 2 минулої лабораторної роботи.

1) Підключення шарів з Geoserver через WFS-сервіс

Шари даних опубліковані на Geoserver можна підключати до створюваної карти через WFS-сервіс. Це дає можливість бачити не лише «картинку» з даними, а й переглядати атрибути об'єктів.

Спочатку необхідно прописати URL-адресу робочого набору Geoserver, де опубліковані дані. Її можна скопіювати з Geoserver, зі сторінки перегляду вмісту шару, вручну змінивши wms на wfs.

```
var url_wfs = 'http://localhost:8080/geoserver/pamyatku/wfs';
```

Після цього прописуємо параметри підключення до потрібних шарів. Оголошуємо **змінну з базовими параметрами** шару та прописуємо *typeName* – назва робочого набору та шару на Geoserver, *maxFeatures* – максимальна кількість одночасно розміщених об'єктів на карті, *srsName* – система координат. Командою **L.Util.extend** перетворюємо задані параметри в рядкову змінну. Створюємо посилання для отримання об'єктів шару комбінуючи URL-адресу з заданими параметрами (отримуючи їх завдяки команді **L.Util.getParamString**).

Командою **L.geoJson.ajax** оголошуємо параметри відображення векторного шару та додаємо його на карту завдяки **addTo**. Основними налаштуваннями тут є стиль відображення (задається параметром *icon* для рисунків умовних знаків точкових об'єктів або параметром *style* для лінійних і площинних об'єктів) та вміст спливаючого вікна, що описується оператором **layer.bindPopup**.

```
var defaultParameters = {
    service: 'WFS',
    version: '2.0.0',
    request: 'GetFeature',
    typeName: 'pamyatku:points_geo',
    maxFeatures: 200,
    outputFormat: 'application/json',
    srsName: 'EPSG:4326'
};
var parameters = L.Util.extend(defaultParameters);
var URL = url_wfs + L.Util.getParamString(parameters);

var points = L.geoJson.ajax(URL, {onEachFeature:function
(feature, layer) {
    layer.bindPopup('<h2>Пам\`ятка</h2><hr/>' +
'Назва: '+feature.properties.pamyatka_n+ '<br>'+ 'Тун: ' +
feature.properties.pamyatka_t);},
    pointToLayer: function(feature, latlng) {
    return L.marker(latlng, {icon: geo})}
}).addTo(мymap);
```

Оголошення параметрів лінійного або площинного шару відрізняється особливістю підключення стилю. У такому випадку назва стилю задається як параметр *style* команди **L.geoJson.ajax**.

```
var defaultParameters1 = {
    service: 'WFS',
    version: '2.0.0',
    request: 'GetFeature',
    typeName: 'pamyatku:UKR_water_lines_dcw',
```

```

    maxFeatures: 200,
    outputFormat: 'application/json',
    srsName: 'EPSG:4326'
  });
  var parameters1 = L.Util.extend(defaultParameters1);
  var URL1 = url_wfs + L.Util.getParamString(parameters1);

  var rivers = L.geoJson.ajax(URL1, { style: vodaStyle,
  onEachFeature: function (feature, layer) {
    layer.bindPopup('<h2> Річка</h2><hr/>' +
    'Назва: '+feature.properties.name);},
    pointToLayer: function(feature, latlng) {
    return L.marker(latlng)}
  }).addTo(map);

```

2) Стили для точкових, лінійних та площинних об'єктів

Для того щоб задати параметри відображення об'єктів на екрані скомпонованої карти необхідно створити та підключити стиль оформлення.

Найпростіший варіант опису стилю точкового об'єкта – оголосити змінну з ім'ям стилю та у вигляді параметрів команди **L.icon** задати *iconUrl* – рисунок з бажаним умовним знаком і його розташування та *iconSize* – розмір рисунка на екрані.

```

var geo = L.icon({
  iconUrl: 'icon/pam.png',
  iconSize: [20, 20]
});

```

Для опису стилю лінійного об'єкта необхідно оголосити змінну з ім'ям стилю та задати параметри *color* – html-кодування кольору, *weight* – товщина лінії, *opacity* – прозорість, *dashArray* – довжина штриха та відстань між штрихами для штрихових ліній.

```

var linStyle = {
  "color": "#000000",
  "weight": 2,

```

```

    "opacity": 1,
    "dashArray": "40 10",
  };

```

Для опису стилю площинного об'єкта необхідно оголосити змінну з ім'ям стилю та задати параметри *color* – html-кодування кольору межі, *weight* – товщина межі, *opacity* – прозорість межі, *fillColor* – html-кодування кольору заливки, *fillOpacity* - прозорість заливки.

```

var plStyle = {
  "color": "#4169E1",
  "weight": 1,
  "opacity": 1,
  "fillColor": "#778899",
  "fillOpacity": 0.5
};

```

Особливості підключення стилів для відображення шару описано вище.

3) Налаштування спливаючих вікон

Як зазначалося вище за вміст спливаючого вікна при WFS-підключенні відповідають параметри оператора **layer.bindPopup**, зокрема після *feature.properties*. необхідно зазначити назву стовпця з якого береться значення атрибуту.

```

layer.bindPopup('<h2>Пам\`ятка</h2><hr/>' + 'Назва: ' +
feature.properties.pamyatka_n + '<br>' + 'Тун: ' +
feature.properties.pamyatka_t);

```

Якщо атрибутивне значення містить посилання на рисунок, можна відобразити рисунок відразу у спливаючому вікні наголосивши, що це джерело зображення *img src* та вказавши його розміри.

```

layer.bindPopup('<h2>Пам\`ятка</h2><hr/>' + 'Назва: ' +
feature.properties.pamyatka_n + '<br>' + 'Тун: ' +
feature.properties.pamyatka_t + '<br>' + '<img src=' +
feature.properties.photo + ' width="200" height="150"/>');

```

Також можна відобразити у спливаючому вікні активні гіперпосилання (рис. 3.1).

```
layer.bindPopup('<h2>' + feature.properties.nazva + '</h2>' + '<hr>' + '<a href="' + feature.properties.opus + '</a>' + '<br>' + '');
```



Рис. 3.1. Спливаюче вікно з активним гіперпосиланням та зображенням

При WFS-підключенні можна налаштувати спливаючі вікна для декількох різних шарів.

4) Налаштування легенди

Загальний принцип підключення легенди описано у питанні 4 минулої лабораторної роботи. Аналогічно групуємо дані та додаємо легенду.

```
var baseLayers = {"OpenStreetMap": osm, "Google": google, "OpenTopoMap": OpenTopoMap };  
var overlays = {"Межі областей": meza, "Населені пункти": nas_punkt, "Озера": lakes, "Річки": rivers, "Пам'ятки": points};
```

```
L.control.layers(baseLayers, overlays, {collapsed:false}).  
addTo(myMap);
```

Для більшої інформативності у легенду можна додати заголовки груп, дописавши їхні назви після опису легенди (рис. 3.2).

```
$( '<h3><b>Перелік базових карт</b></h3><hr  
>').insertBefore('div.leaflet-control-layers-base');  
$( '<h3><b>Перелік векторних шарів</b></h3><hr  
>').insertBefore('div.leaflet-control-layers-overlays');
```

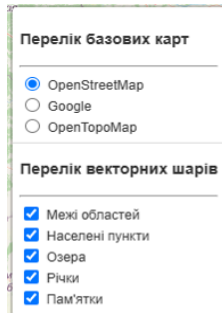


Рис. 3.2. Легенда із заголовками груп шарів

Також для кожного векторного шару у легенді можна додати відображення використаного умовного знаку. Необхідно створити рисунки з фрагментами використаних умовних знаків та відобразити їх у легенді.

```
var baseLayers = {  
    "OpenStreetMap": osm,  
    "Google": google,  
    "OpenТopoMap": OpenТopoMap};  
var overlays = {  
    "Межі областей": meza,  
    "Населені пункти": nas_punkt,  
    "Озера": lakes,  
    "<img height='15' width='25'  
    src='img/riv_uz.png'/> <span class='my-layer-  
    item'>Річку</span>": rivers,  
    "<img height='15' width='15' src='img/pam.png'/>
```



```
<span class='my-layer-item'>Пам'ятки</span>":  
points};
```

```
L.control.layers(baseLayers, overlays, {collapsed:false}).  
addTo(mymap);
```

```
$( '<h3><b>Перелік базових карт</b></h3><hr  
>').insertBefore('div.leaflet-control-layers-base');  
$( '<h3><b>Перелік векторних шарів</b></h3><hr  
>').insertBefore('div.leaflet-control-layers-overlays');
```

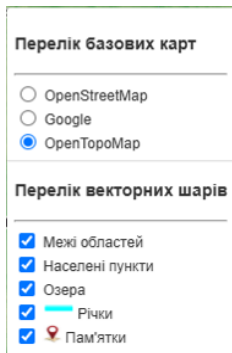


Рис. 3.3. Легенда із відображенням використаних умовних знаків

5) Додаткові панелі інструментів

Для розширення функціональності створеної карти можна розмістити додаткові інструменти, наприклад: геолокацію, картометричні інструменти, інструменти пошуку тощо. Це здійснюється через використання плагінів, з переліком яких можна ознайомитися на сторінці <https://leafletjs.com/plugins.html>. Крім того додатковий інструментарій можна шукати на спеціалізованих сайтах типу GitHub <https://github.com/>.

Кожеу плагін може мати власні особливості встановлення, з якими можна ознайомитися на його сторінці. Однак, в загальному послідовність наступна: необхідно дописати у html документ посилання на стиль оформлення панелі, посилання на бібліотеку та власне прописати виклик елемента керування.

Визначення місцеположення

У тезі `<head>` додаємо посилання на стиль оформлення панелі геолокації.

```
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/leaflet.locatecontrol/dist/L.Control.Locate.min.css" />
```

У тезі `<body>` додаємо скрипт з підключенням відповідної js-бібліотеки.

```
<script
src="https://cdn.jsdelivr.net/npm/leaflet.locatecontrol/dist/L.Control.Locate.min.js" charset="utf-8"></script>
```

У основному скрипті з компонуванням карти, після оголошення карти та підключення бібліотек додаємо елемент керування (рис. 3.4).

```
var lc = L.control.locate({
  position: 'topleft',
  strings: {title: "Покажи мені де я"}
}).addTo(map);
```

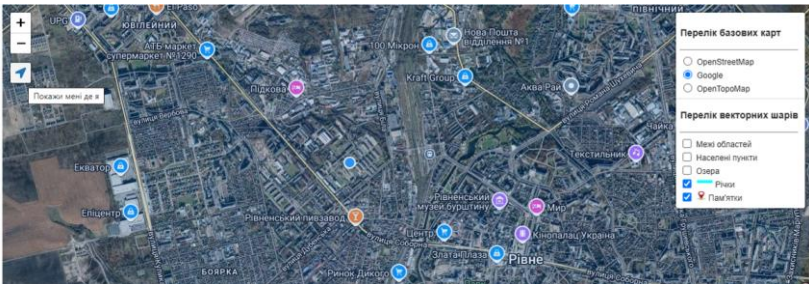


Рис. 3.4. Робота інструмента визначення місцеположення

Вимірювання відстані

У тезі `<head>` додаємо посилання на стилі оформлення панелі вимірювання відстані.

```
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/leaflet.draw/dist/leaflet.draw.min.css" />
```

```
<link rel="stylesheet" href="Leaflet.MeasureControl-  
master/leaflet.measurecontrol.css"/>
```

У тезі `<body>` додаємо скрипти з підключенням відповідних js-бібліотек.

```
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/leaflet.draw/1.0.4/  
eaflet.draw.js"></script>
```

```
<script src="Leaflet.MeasureControl-  
master/leaflet.measurecontrol.js"></script>
```

Якщо стилі чи бібліотеки містять посилання на локальні файли, то обов'язково додаємо їх у відповідну папку!!! У даному прикладі відбувається підключення стилів та бібліотек не лише з інтернету, а й з локальної папки *Leaflet.MeasureControl-master*. Вміст папки з необхідними файлами можна завантажити на сторінці відповідного плагіна <https://leafletjs.com/plugins.html#measurement>.

У основному скрипті з компонуванням карти, після оголошення карти та підключення бібліотек додаємо елемент керування (рис. 3.5).

```
L.Control.measureControl({position: 'topleft'}).addTo(myMap);
```



Рис. 3.5. Робота інструмента вимірювання відстані

Пошук за назвою адміністративної одиниці

У тезі `<head>` додаємо посилання на стиль оформлення панелі пошуку.

```
<link rel="stylesheet" href="src/leaflet-search.css" />
```

У тезі <body> додаємо скрипт з підключенням відповідної js-бібліотеки.

```
<script src="src/leaflet-search.js"></script>
```

У даному прикладі відбувається підключення стилів та бібліотек з локальної папки *src*. Вміст папки з необхідними файлами можна завантажити на сторінці відповідного плагіна <https://leafletjs.com/plugins.html#search--popups>.

У основному скрипті з компонуванням карти, після оголошення карти та підключення бібліотек додаємо елемент керування (рис. 3.6).

```
map.addControl( new L.Control.Search({  
    url:  
    'https://nominatim.openstreetmap.org/search?format  
    =json&q={s}',  
    jsonParam: 'json_callback',  
    propertyName: 'display_name',  
    propertyLoc: ['lat','lon'],  
    marker: L.circleMarker([0,0],{radius:0}),  
    autoCollapse: true,  
    autoType: false,  
    minLength: 2  
}));
```

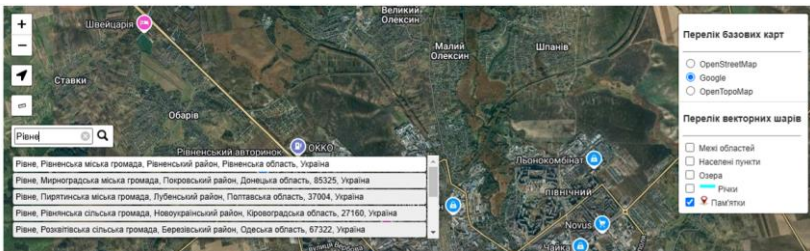


Рис. 3.6. Робота інструмента пошуку адміністративної одиниці

Додаткові плагіни

Підібрати плагін для своїх потреб можна на сторінці <https://leafletjs.com/plugins.html>. Також можна рекомендувати спробувати наведені нижче додаткові плагіни.

Створення «шторки» для порівняння карт
<https://github.com/digidem/leaflet-side-by-side>

Синхронізація карт (декілька карт-врізок на екрані, які одночасно змінюють масштаб)

<https://github.com/jieter/Leaflet.Sync>

Відображення переліку базових карт у вигляді «іконок»
<https://github.com/ScanEx/Leaflet-IconLayers>

Перегляд фрагменту карти через «збільшувальне скло»
<https://github.com/bbecquet/Leaflet.MagnifyingGlass>

Невелика міні-карта, що показує загальну карту в дрібнішому масштабі для полегшення навігації.

<https://github.com/Norkart/Leaflet-MiniMap>

Завдання для самостійної роботи. Розглянути можливості бібліотеки GeoExt для компонування WFS карти. Спробувати підключити інші плагіни, не розглянуті у цих методичних вказівках.

Запитання для контролю:

1. Як підключити шар з Geoserver як WFS-сервіс?
2. Яка перевага підключення через WFS?
3. Які налаштування шару задаються при підключенні через WFS?
4. Як задати вміст спливаючого вікна при WFS-підключенні?
5. Як додати відображення рисунка у спливаюче вікно?
6. Як додати активне гіперпосилання у спливаюче вікно?
7. Як додати стиль для лінійного об'єкта?
8. Як додати стиль для точкового об'єкта?
9. Як підключити базові карти?
10. Як додати умовні позначення у легенду?
11. Як згрупувати шари у легенді та відобразити їх заголовки?
12. Як додати панель геолокації?
13. Як додати інструмент для вимірювання відстані?
14. Як додати інструмент для пошуку населеного пункту за назвою?

РЕКОМЕНДОВАНА ЛІТЕРАТУРА ТА РЕСУРСИ

1. Геоінформаційні системи і бази даних : монографія / В. І. Зацерковний, В. Г. Бурачек, О. О. Железняк, А. О. Терещенко. Ніжин : НДУ ім. М. Гоголя, 2014. 492 с.
2. Геоінформаційні технології та інфраструктура геопросторових даних: у шести томах. Том 2: Системи керування базами геоданих для інфраструктури просторових даних : навчальний посібник / Кейк Д., Лященко А. А., Путренко В. В., Хмелевський Ю., Дорошенко К. С., Говоров М. К. : Планета-Прінт, 2017. 456 с.
3. Основи створення інтегрованих геопросторових даних. / Ю. О. Карпінський та ін. Київ : КНУБА, 2023. 302 с.
4. GeoServer. URL:<https://geoserver.org/>
5. GitHub Discussions. The home for developer communities. URL: <https://github.com/features/discussions>
6. Leaflet - a JavaScript library for interactive maps. URL: <https://leafletjs.com/>
7. Leaflet Plugins database URL:<https://leafletjs.com/plugins.html>
8. Open Geospatial Consortium. Standards. Simple Feature Access – Part 1: Common Architecture. URL: <http://www.opengeospatial.org/standards/sfa>
9. QGIS - провідна вільна настільна ГІС. URL: <https://www.qgis.org/uk/site/about/index.html>
10. SQL. URL: <https://uk.wikipedia.org/wiki/SQL>
11. Web Feature Service (WFS) Standard. OGC Publications. URL: <https://www.ogc.org/publications/standard/wfs/>
12. Web Map Service (WMS) Standard | OGC Publications. URL: <https://www.ogc.org/publications/standard/wms/>