



support

Materiały szkoleniowe
WebGIS - Geoserver, PostGIS, QGIS, OpenLayers
(poziom podstawowy)



**MINISTERSTWO
KLIMATU**



Sfinansowano ze środków
Narodowego Funduszu
Ochrony Środowiska
i Gospodarki Wodnej

Spis treści:

Architektura aplikacji internetowej	3
Elementy systemów sieciowych	3
Front-end	3
Back-end	5
Architektura klient-serwer	5
Narzędzia deweloperskie przeglądarki	7
Protokół HTTP(S)	8
Metody HTTP	9
Nagłówki HTTP	10
Kod odpowiedzi HTTP	10
Ciasteczka	11
Specyfika systemów webgis	11
Otwarte standardy	11
Dane przestrzenne w systemach webgis	12
Baza danych przestrzennych PostGIS	16
Instalacja w systemie Windows	16
pgAdmin	28
PostGIS	32
QGIS i PostGIS	34
Łączenie z bazą danych	34
DB Manager	35
Import danych wektorowych do bazy PostGIS	36
Publikowanie danych w aplikacji Geoserver	42
Instalacja w systemie Windows	43
Panel administracyjny	43
Zarządzanie danymi przestrzennymi	45
Obszary robocze (workspaces)	46
Magazyny danych (stores)	49
Warstwy (layers)	57
Podgląd warstw	65
Publikacja danych	65
Wczytanie danych do QGIS	66
WMS/WMTS	66
WFS	67
Stylizacja	68
Symbolizacja danych w QGIS	70
Przypisanie stylu do warstwy	74
Tworzenie serwisów internetowych	78

Serwer WWW	78
Konfiguracja nginx	80
Pliki statyczne	81
Serwer pośredniczący	83
OpenLayers	84
Instalacja biblioteki	85
Dodanie mapy na stronie internetowej	86
Widok mapy	86
Dodawanie warstw	87
Rastrowe warstwy kaflowe	87
WMS	89
Dane wektorowe	93
Rozszerzenia OpenLayers	96
Legenda mapy	97

Architektura aplikacji internetowej

Aplikacja internetowa (webowa, *web application*) to program zainstalowany na serwerze, który pozwala na komunikację z aplikacją kliencką (zazwyczaj przeglądarka internetowa). Przeglądarka wysyła do serwera na podany adres URL żądanie (np. pobranie strony lub pliku), aplikacja internetowa przetwarza je i zwraca odpowiedź (np. plik HTML, obraz).

Systemy webgis pozwalają na zarządzanie danymi przestrzennymi z poziomu przeglądarki. Użytkownik może m.in. wyświetlić dane w formie mapy, filtrować informacje, tworzyć nowe i edytować istniejące elementy. Dzięki dużemu rozwojowi technologii internetowych (m.in. *JavaScript*) systemy tego typu mają coraz większe możliwości i w dużej mierze potrafią zastąpić rozwiązania desktopowe. Zaletą webgis dla użytkowników końcowych jest brak konieczności instalowania dodatkowego oprogramowania, najczęściej wystarczy przeglądarka. Umożliwiają one również wykonywanie skomplikowanych obliczeń w znacznie krótszym czasie, ponieważ często serwery mają większą moc obliczeniową niż komputery stacjonarne i laptopy lub mogą wykonywać obliczenia w systemach rozproszonych (są one wykonywane równolegle na wielu komputerach). Duże znaczenie ma również możliwość centralizacji dostępu do danych co ułatwia m.in. równoczesną pracę wielu osób, szybkie przesyłanie wyników pomiędzy użytkownikami i łatwiejszą kontrolę dostępu do informacji oraz łatwiejszy proces aktualizacji oprogramowania.

Należy jednak pamiętać, że dostęp do aplikacji webowych wymaga stałego połączenia internetowego oraz serwera. W razie przerwania połączenia sieciowego pełne wykorzystanie ich możliwości jest utrudnione lub wręcz niemożliwe. Sam serwer również może ulec awarii co grozi utratą danych. Dlatego tak ważnym zagadnieniem jest regularne tworzenie ich kopii bezpieczeństwa (tzw. *backup*). Również pomimo rozwoju technologii internetowych aplikacje często są mniej responsywne od ich desktopowych odpowiedników, szczególnie przy większych zbiorach danych.

Elementy systemów sieciowych

Aplikacje internetowe składają się z dwóch głównych komponentów:

- **front-end** - warstwa prezentująca informacje i pozwalająca na interakcję z użytkownikiem, dane pobierane są z backendu za pomocą np. protokołu HTTP; najczęściej jest to strona internetowa wyświetlana w przeglądarce.
- **back-end** - część serwerowa, niewidoczna dla użytkownika, obsługuje zapytania HTTP, przechowuje i zarządza danymi itp.

Front-end

W przypadku stron WWW front-end jest stroną internetową, która jest wyświetlana w przeglądarce (aplikacja kliencka). Jest to interfejs, dzięki któremu użytkownik komunikuje się z serwerem (back-end).

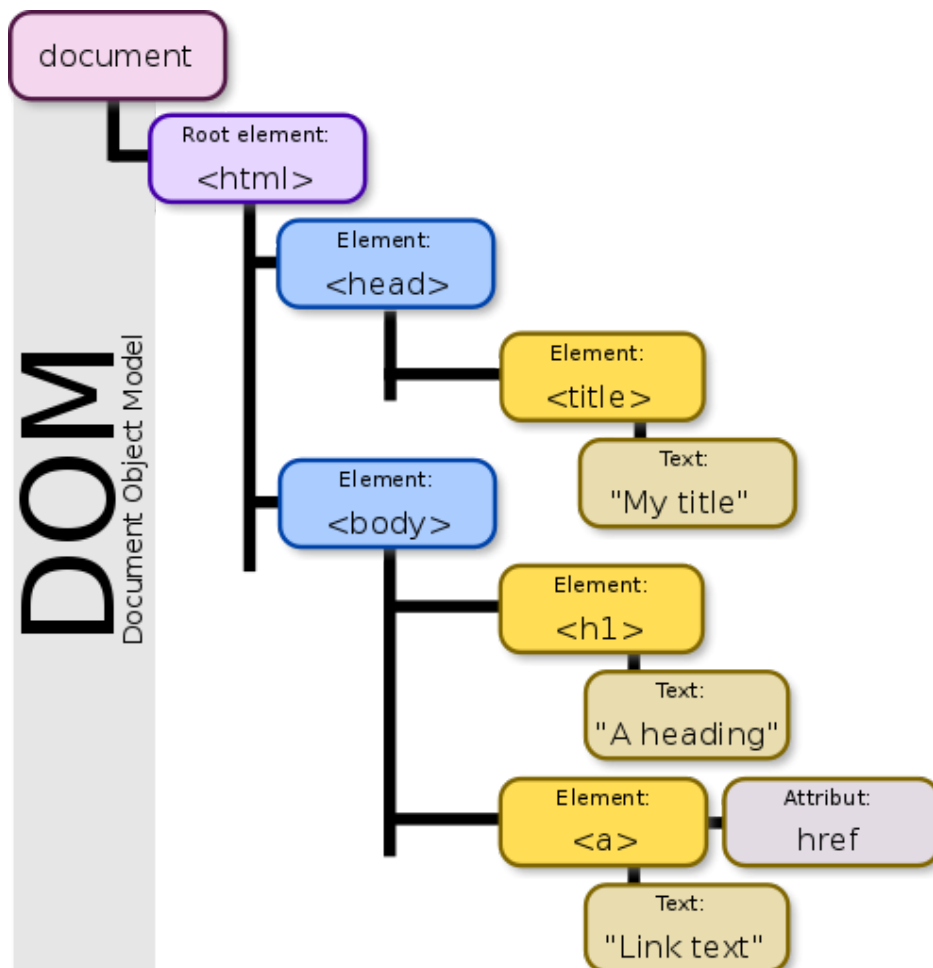
Języki programowania wykorzystywane do tworzenia stron internetowych:

- **HyperText Markup Language (HTML)** - opisuje strukturę strony WWW i jej elementów,

- **Cascading Style Sheets (CSS)** - język pozwalający zarządzać wyglądem (stylami) elementów strony WWW,
- **JavaScript (JS)** - pozwala na sterowanie elementami strony, wysyłanie żądań, interakcję z użytkownikiem czy wykonywanie operacji obliczeniowych bez obciążania serwera. Od 1997 r. oficjalną nazwą jest ECMAScript.

W skrócie można stwierdzić, że język znaczników HTML definiuje strukturę strony, CSS służy do stylizacji poszczególnych elementów, a JavaScript pozwala manipulować elementami dokumentu HTML i wykonywać obliczenia.

Strony internetowe pisane w języku HTML mają strukturę hierarchiczną. Budowana ona jest za pomocą znaczników (tagów) definiujących konkretne elementy strony, mogą one być zagnieżdżane tworząc strukturę drzewa. Model reprezentujący stronę w formie powiązanych ze sobą obiektów o usystematyzowanej strukturze określa się jako HTML DOM (*Document Object Model*). Każdy typ obiektu DOM posiada własne właściwości, metody i zdarzenia.



Źródło: Wikipedia, https://en.wikipedia.org/wiki/Document_Object_Model

Za pomocą JavaScript możliwe jest pobieranie, zmienianie, dodawanie i usuwanie elementów HTML oraz reagowanie na związane z nimi zdarzenia, dzięki czemu strony reagują na działania użytkownika bez konieczności ich przeładowania. Ponadto możliwe jest wykonywanie obliczeń, dzięki czemu serwer nie jest obciążony.

Back-end

Back-end znajduje się na jednym lub więcej serwerów, czyli komputerów podłączonych i skonfigurowanych do łączenia się z siecią. Jest na nim zainstalowane specjalne oprogramowanie umożliwiające komunikację z innymi komputerami. Najczęściej aplikacje te można podzielić na 3 kategorie:

- Serwer WWW (lub Serwer HTTP) służy do obsługi komunikacji za pośrednictwem protokołu HTTP(S),
- Aplikacja internetowa (webowa) to dedykowany program przetwarzający żądania HTTP i zwracający odpowiedzi,
- Baza danych lub inne źródło informacji - miejsce przechowywania danych.

Serwery WWW pozwalają administratorowi na skonfigurowanie dostępu do serwera z zewnątrz. Umożliwiają m.in. przesyłanie plików z dysku lub korzystanie z innego oprogramowania (aplikacji webowej) do zwracania odpowiedzi. Najpopularniejszymi serwerami WWW są *Apache* i *nginx*, wg różnych szacunków obsługują od 60 do 90% całego ruchu internetowego na świecie. Oba są projektami Open Source i mogą być wykorzystywane bez opłat również w celach komercyjnych. Serwery WWW mogą zwracać pliki statyczne zlokalizowane na dysku, w takim wypadku nie ma potrzeby instalowania dodatkowych aplikacji do obsługi serwisów internetowych.

Aplikacja webowa to dodatkowe oprogramowanie do przetwarzania danych. Pozwala tworzyć bardzo zaawansowane, dynamiczne serwisy internetowe. Najczęściej są ukryte za serwerem WWW, który służy jako tzw. serwer pośredniczący (*proxy server*) czyli jedynie przekazuje żądania i odpowiedzi z/do właściwej aplikacji internetowej.

Języki programowania wykorzystywane do tworzenia aplikacji webowych to m.in.: *Python*, *PHP*, *Ruby*, *Java*, *JavaScript* (dzięki środowisku *node.js*), *C++* i wiele innych.

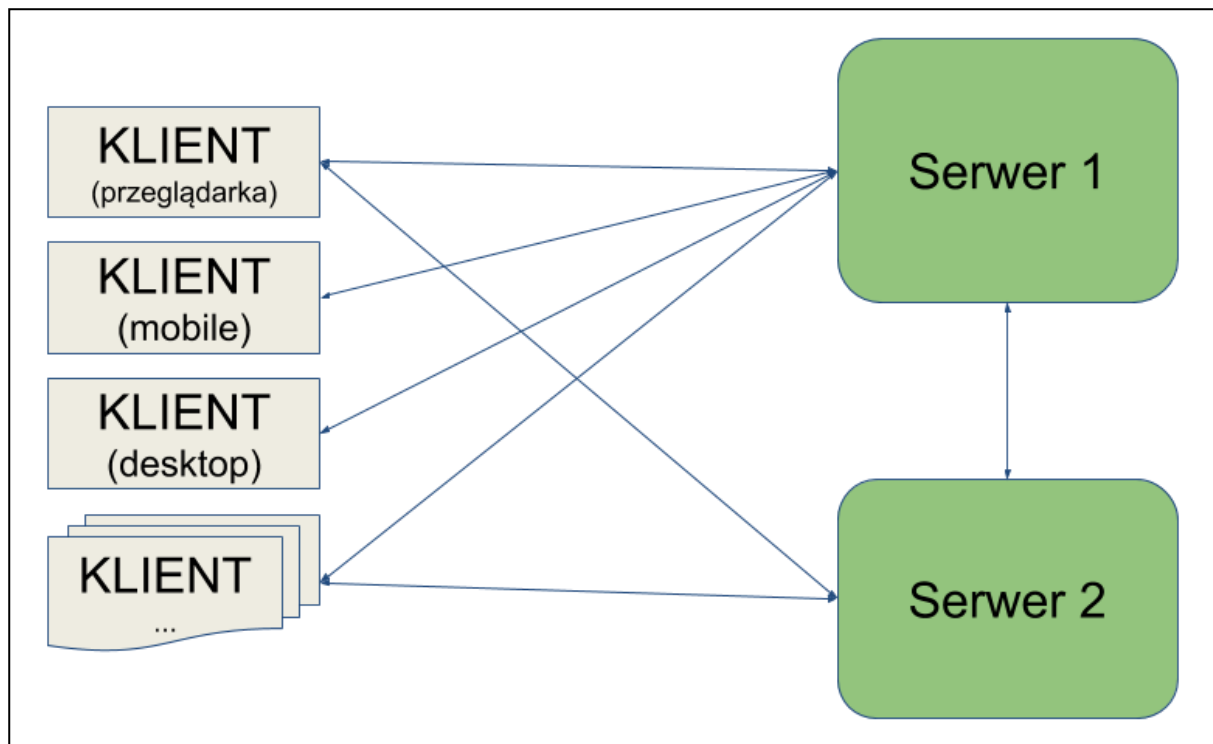
Dane przechowywane są zazwyczaj albo w formie plików na dysku (szczególnie dane binarne) lub w bazach danych. To drugie rozwiązanie pozwala przechowywać dane w odpowiedniej strukturze, z uwzględnieniem relacji pomiędzy różnymi obiektami. Umożliwiają one również przetwarzanie i analizowanie zgromadzonych danych.

Najpopularniejszym typem są tzw. relacyjne bazy danych (RDBMS - system zarządzania relacyjną bazą danych). Istnieje wiele aplikacji bazodanowych zarówno komercyjnych (*Oracle*, *SQL Server*, *Access*) jak i dostępnych na wolnych licencjach (*PostgreSQL*, *SQLite*, *MySQL*).

Architektura klient-serwer

Architektura klient-serwer opisuje ogólny sposób działania rozproszonych systemów, w których role są podzielone na klienta i serwer. Klientem określa się aplikację która żąda zasobu lub dostępu do usługi. Może to być przeglądarka internetowa, aplikacja mobilna lub desktop. Serwer natomiast jest stroną świadczącą usługi i udostępniającą zasoby. W ogólnym ujęciu klient wysyła żądanie do serwera, ten je przetwarza i zwraca wynik.

Jedna aplikacja kliencka może komunikować się równolegle z wieloma serwerami, a jeden serwer może obsługiwać jednocześnie wiele zapytań od aplikacji klienckich:



W Internecie komunikacja odbywa się za pomocą *protokołu kontroli transmisji* (TCP). Każdy komputer podłączony do sieci posiada unikalny identyfikator tzw. adres IP (*Internet Protocol address*). Jest to zbiór liczb, dzięki którym możliwe jest przekazywanie informacji pomiędzy oddalonymi komputerami. Podstawową wersją jest IPv4 zapisujący adres w formie 4 liczb z zakresu 0-255. Ze względu na znaczny przyrost urządzeń podłączonych do sieci (głównie ze względu na rozwój tzw. *IoT, Internet of thing* - Internet rzeczy) stworzono następcę IPv6. Aby uprościć dostęp do serwerów korzysta się z domen, które są zarejestrowane i wskazują na adres IP. Np. domena *gov.pl* wskazuje na adres IP 194.181.92.99, a *gis-support.pl* ma adres IP 185.135.90.37. Za tłumaczenie domen na adresy IP odpowiada system nazw domen (*DNS, Domain Name System*).

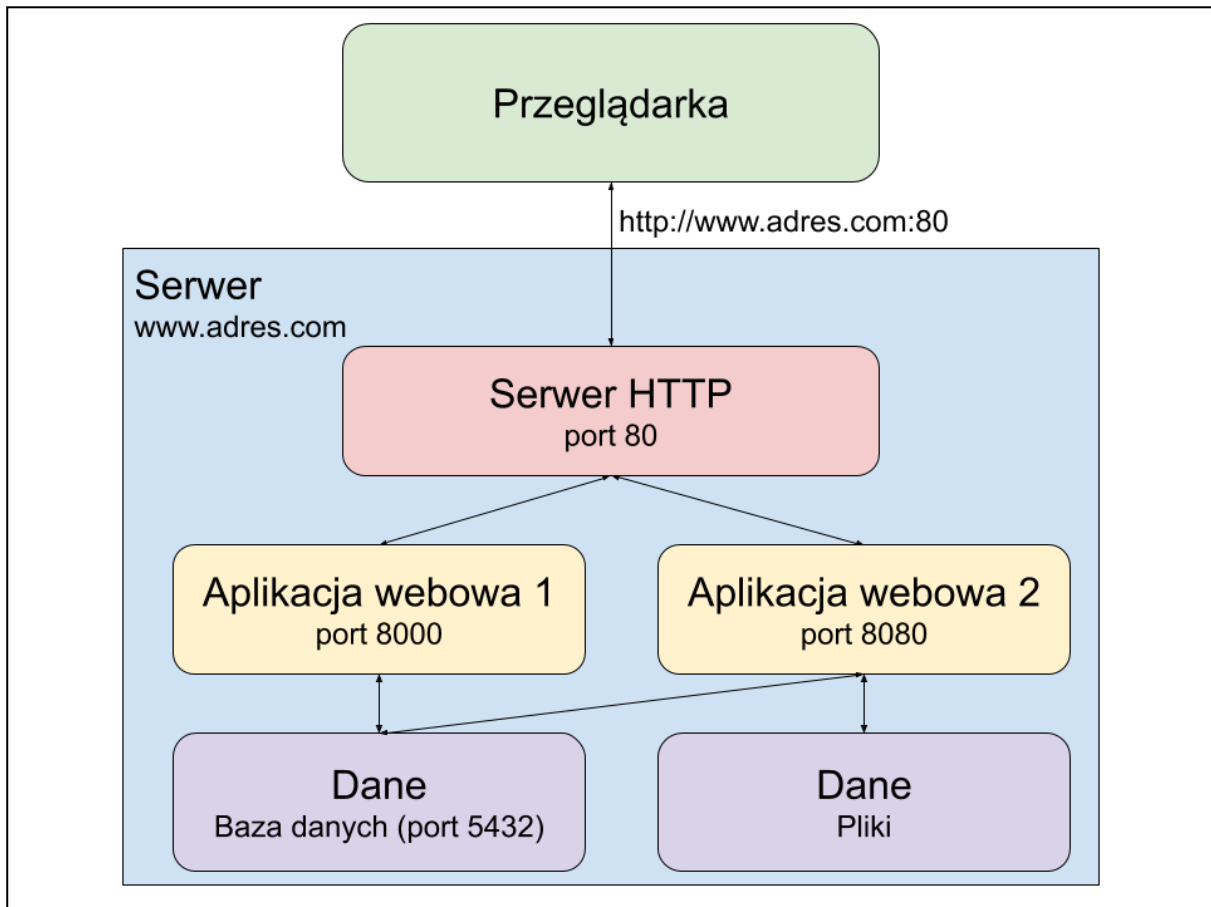
Część adresów IP jest zarezerwowana pod konkretne zastosowania. Przykładem jest adres 127.0.0.1 (*localhost*), dzięki któremu komputer może połączyć się z samym sobą. W tym przypadku żądanie nie jest przesyłane na zewnątrz tylko przetwarzane na tym samym komputerze.

W danym momencie może być uruchomionych wiele procesów, które obsługują różne typy żądań. Np. może być uruchomiony serwer WWW, serwer poczty, FTP i baza danych. Aby system wiedział do którego procesu ma przekazać dane żądanie każdy z nich ma przypisany unikalny identyfikator tj. port protokołu. Jest to liczba naturalna z zakresu od 0 do 65535. Port jest określany przez klienta razem z adresem IP/domeną po dwukropku np. 194.181.92.99:443 lub *gov.pl:443*. niektóre numery portów są określone jako główne i powiązane z konkretnymi typami usług. W takich wypadkach można je pominąć przy podawaniu adresu w przeglądarce, doda ona go sama. Wybrane porty i ich zastosowanie:

- 80 - HTTP
- 443 - HTTPS

- 5432 - PostgreSQL

Przykład komunikacji elementów aplikacji internetowych w architekturze klient-serwer z wykorzystaniem domeny i portu:



Na powyższym przykładzie przeglądarka komunikuje się z serwerem za pomocą protokołu HTTP. Serwer WWW na podstawie zdefiniowanych reguł decyduje, do której aplikacji webowej zostanie przekazane żądanie.

Narzędzia deweloperskie przeglądarki

Większość przeglądarek udostępnia zestaw narzędzi ułatwiający tworzenie i analizowanie stron WWW. Pozwalają one programistom na śledzenie działania strony w celu zweryfikowania czy działa ona poprawnie.

W przeglądarkach *Chrome* i *Firefox* dostęp do narzędzi deweloperskich można uzyskać:

- klikając prawym klawiszem na stronie i wybierając opcję *Zbadaj* (Chrome) lub *Zbadaj element* (Firefox),
- wybierając skrót klawiaturowy Ctrl+Shift+I (jak Irena).

W trakcie szkolenia wykorzystywana będzie przeglądarka *Firefox*, jednak omawiane narzędzia są dostępne również m.in. w *Chrome*.

Poszczególne narzędzia pogrupowane są w zakładkach, nazwy w poszczególnych przeglądarkach mogą nieznacznie się różnić, ale w podstawowe funkcjonalności są do siebie bardzo podobne. Najważniejsze zakładki to:

- **Inspektor (Firefox) / Elements (Chrome)** - analiza struktury strony (HTML DOM),

- **Sieć (Firefox) / Network (Chrome)** - listowanie zapytań wychodzących z przeglądarki dla danej strony,
- **Konsola (Firefox) / Console (Chrome)** - interaktywna konsola JavaScript, wywołanie w kodzie JS polecenia `console.log(obiekt)` pozwala wydrukować dowolny obiekt w konsoli przeglądarki np. tekst, liczbę, tablicę.

Przykładowy widok zapytań wykonywanych przez przeglądarkę podczas wczytania pojedynczej strony internetowej. Po kliknięciu w element listy pojawia się panel z jego szczegółami.

The screenshot shows the Chrome DevTools Network tab. The top part displays a list of requests:

Stan	M...	Domena	Plik	Inicj...	Typ	Przesłano	Rozmiar
200	GET	gis-support.pl	/	doc...	ht...	13,54 KB	70,51 KB
200	GET	gis-support.pl	jquery.js?ve	script	js	w pamięci podręc...	94,60 KB
200	GET	gis-support.pl	jquery.prett	script	js	w pamięci podręc...	21,47 KB
200	GET	gis-support.pl	video-lightl	script	js	w pamięci podręc...	6,76 KB
200	GET	gis-support.pl	jquery.prett	script	js	w pamięci podręc...	22,35 KB

Below the list, the details for the selected GET request are shown:

46 żądań | Przesłano: 4,44 MB / 600,88 KB | 12,13 s | DOMContentLoaded: 834 ms | load: 1,87 s

Nagłówki | Ciasteczka | Żądanie | Odpowiedź | Pomiar czasu | Bezpieczeństwo

Filtruj nagłówki | Blokuj | Wyślij ponownie

▶ GET https://gis-support.pl/

Stan: 200 OK ?

Wersja: HTTP/1.1

Przesłano: 13,54 KB (o rozmiarze 70,51 KB)

Zasada polecającego: no-referrer-when-downgrade

▼ Nagłówki odpowiedzi (486 B) Nieprzetworzone

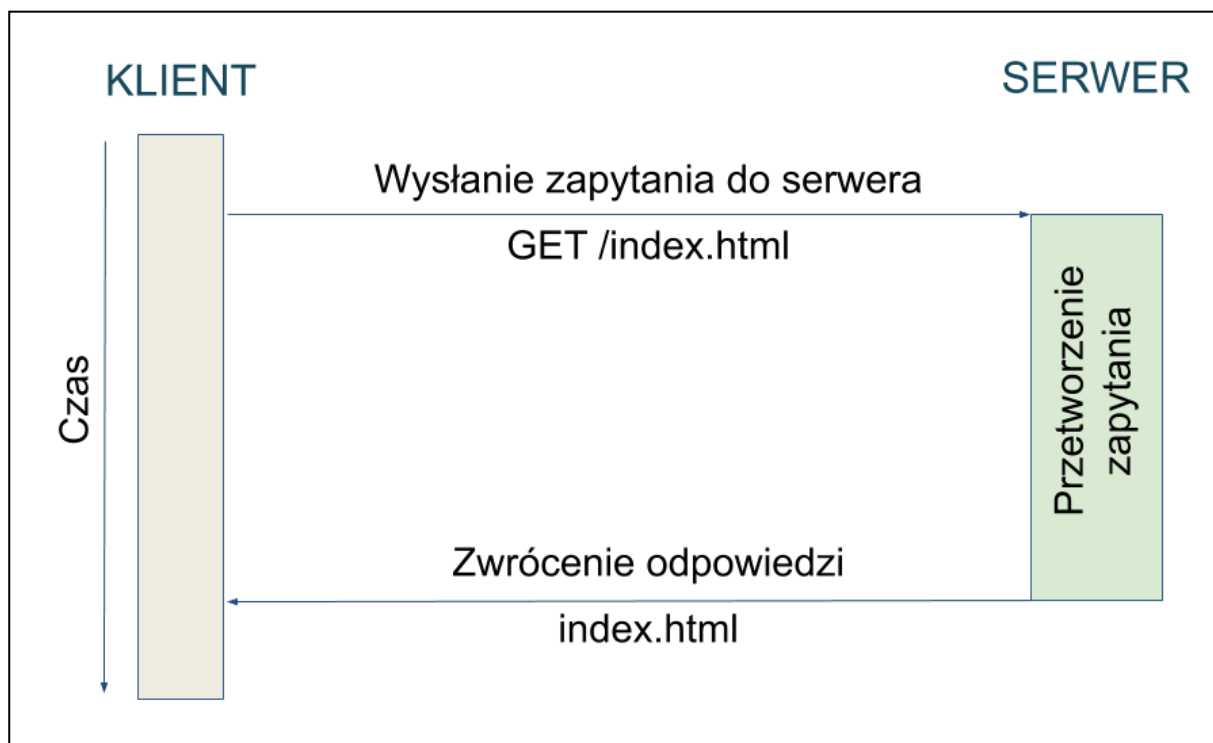
- Accept-Ranges: none
- Connection: Keep-Alive
- Content-Encoding: gzip
- Content-Length: 13382
- Content-Type: text/html; charset=UTF-8
- Date: Tue, 08 Sep 2020 09:44:01 GMT
- Keep-Alive: timeout=5, max=93
- Link: <https://gis-support.pl/wp-json/>; rel="https://api.w.org/", <https://gis-support.pl/wp-json

Protokół HTTP(S)

Protokół HTTP (*Hypertext Transfer Protocol*) służy do przesyłania dokumentów hipertekstowych np. stron WWW oraz danych. Na każde zapytanie składa się zestaw informacji niezbędnych do ich przesłania i realizacji po stronie serwera.

HTTP to protokół, w którym informacje są przesyłane za pomocą tekstu. HTTPS (nazywany czasem *HTTP przez SSL*) natomiast pozwala szyfrować przesyłane informacje za pomocą różnych protokołów (w przeszłości SSL, obecnie TLS). Zapobiega to przechwytywaniu i zmienianiu przesyłanych danych. Szyfrowanie wymaga posiadanie certyfikatu klucza publicznego, który jest podpisany przez tzw. zaufaną trzecią stronę, czyli podmiot uprawniony do wystawiania cyfrowych certyfikatów. Większość z nich prowadzi działalność komercyjną, możliwe jest jednak zdobycie certyfikatu darmowego na stronie <https://letsencrypt.org/>.

HTTP(S) jest bezstanowe - każde kolejne żądanie jest niezależne od poprzednich, żadne informacje nie są automatycznie zachowywane. Istnieją techniki pozwalające na zapamiętywanie tych informacji i dołączanie do kolejnych transakcji (np. danych zalogowanego użytkownika) m.in. ciasteczka (cookies), sesje serwerowe, dodatkowe parametry w URL.



Każde żądanie opisane jest metadanymi na które składają się metody HTTP, nagłówki, ciasteczka i kody odpowiedzi.

Metody HTTP

Metody wysyłane są do konkretnego zasobu identyfikowanego za pomocą adresu URL. Określają one czynność jaką chcemy wykonać na danym zasobie i sposób przekazywania dodatkowych informacji. Są wysyłane wraz z żądaniem przez klienta.

- **GET** - pobranie zasobu z serwera, dodatkowe parametry można przekazać w URL,
- **POST** - wysłanie danych do serwera, najczęściej wykorzystywana do logowania i wysyłania danych z formularzy ponieważ przesyłane dane są ukryte i mogą być zaszyfrowane (w przypadku korzystania z HTTPS),

- **PUT** - wysłanie danych do serwera w celu utworzenia lub aktualizacji zasobu, w przeciwieństwie do POST metoda jest idempotentna tzn. wielokrotne wysłanie żądania ma zawsze taki sam skutek (np. nowy zasób nie jest tworzony jeśli istnieje)
- **DELETE** - usunięcie zasobu.

Inne często spotykane metody to m.in. PATH, OPTIONS i HEAD. Dzięki różnym metodom ten sam adres URL może służyć do obsługi różnych czynności np. pobrania (GET) i dodania nowego elementu (POST).

Nagłówki HTTP

Nagłówki to dodatkowe dane przesyłane wraz z zapytaniem i odpowiedzią w formie `nazwa:wartość`. Można o nich myśleć jako metadanych żądania. Przykładowe nagłówki:

- **Referer** - adres strony, z której wysłano zapytanie
- **User-Agent** - identyfikator aplikacji wysyłającej zapytanie
- **Accept** - akceptowane typy MIME odpowiedzi (format zwracanych danych)
- **Cookie** - treść ciasteczka przesyłana do serwera
- **Content-Type** - format MIME odpowiedzi (odpowiednik *Accept*)
- **Content-Length** - długość odpowiedzi
- **Date** - data wygenerowania odpowiedzi

Możliwe jest przesyłanie również nagłówków, które nie są opisane w specyfikacji HTTP, ich nazwę należy rozpocząć od X- np. *X-Update-Date*.

Kod odpowiedzi HTTP

Każda odpowiedź na żądanie do serwera zwraca informację o statusie w formie trzycyfrowego kodu. Pierwsza cyfra informuje o sposobie realizacji zapytania:

- **1XX** - kody informacyjne
- **2XX** - kody powodzenia
- **3XX** - kody przekierowania
- **4XX** - kody błędu aplikacji klienta
- **5XX** - kody błędu serwera HTTP

Wybrane kody odpowiedzi:

- **200 OK** - żądanie się powiodło, np. dla GET pobrano zasób,
- **201 Created** - informuje o poprawnym utworzeniu zasobu np. dla POST udana rejestracja nowego użytkownika w serwisie,
- **301 Moved Permanently** - Informuje o przeniesieniu zasoby pod inny URL, nowy adres jest zwracany z odpowiedzią,
- **400 Bad Request** - wysłano niepoprawne dane i serwer nie może zwrócić odpowiedzi np. niepoprawne dane logowania,
- **404 Not Found** - nie znaleziono zasobu pod wskazanym URL np. strona nie istnieje,
- **500 Internal Server Error** - błąd po stronie serwera, np. błąd w aplikacji przetwarzającej żądanie.

Ciasteczka

Ciasteczka są tekstem, który może być dołączany do żądań i odpowiedzi. Używane są głównie do informowania serwera o użytkowniku np. po zalogowaniu pozwalają przeglądać witrynę bez konieczności przesyłania loginu i hasła przy każdym żądaniu. Dzięki temu użytkownik nie musi wpisywać tych samych informacji za każdym razem, gdy powróci na tę stronę lub przejdzie z jednej strony na inną.

Ciasteczka mogą mieć postać zaszyfrowaną, jeśli dany serwer na to pozwala.

Specyfika systemów webgis

Tworzenie rozbudowanych aplikacji internetowych wymaga wykorzystania wielu komponentów. Na rynku dostępna jest szeroka gama oprogramowania do realizowania wybranych funkcjonalności, zarówno komercyjnych jak i Open Source. Często rozwiązaniem jest tworzenie systemów hybrydowych, w których część funkcjonalności realizowanych jest przez otwarte projekty, a część przez programy komercyjne. Dzięki temu można zbilansować koszty wdrożenia i utrzymania oprogramowania przy zachowaniu optymalnych dla użytkowników możliwości. W ramach szkolenia skoncentrujemy się na rozwiązaniach Open Source, dlatego w dalszej części zostaną omówione tego typu aplikacje.

Istnieje bardzo dużo projektów Open Source związanych zarówno z danymi przestrzennymi jak i tworzeniem aplikacji internetowych. Dzięki temu możliwe jest stworzenie dużych i zaawansowanych systemów GIS tylko w oparciu o otwarte rozwiązania. Aplikacje komercyjne również mogą korzystać z otwartych projektów i standardów.

Rozwój wielu projektów Open Source GIS wspierany jest przez pozarządową organizację non-profit *Open Source Geospatial Foundation (OSGeo)*. Pomaga ona nie tylko w rozwoju wolnego oprogramowania ale również promuje otwarte standardy oraz prowadzi działalność edukacyjną i szkoleniową m.in. poprzez organizowanie corocznej konferencji *Free and Open Source Software for Geospatial (FOSS4G)*. Wśród projektów rozwijanych w ramach OSGeo są m.in. biblioteka *GDAL/OGR*, *PostGIS*, *QGIS*, *Geoserver*, *OpenLayers*.

Na systemy Windows udostępniany jest specjalny program *OSGeo4W* ułatwiający instalację wielu aplikacji GIS rozwijanych w ramach tej fundacji.

Otwarte standardy

Otwarte standardy pozwalają na dostęp do pełnej specyfikacji danych zagadnień i ich implementację w dowolnym oprogramowaniu. Nie są one limitowane prawnie i finansowo, dzięki czemu mogą być stosowane bez ograniczeń w aplikacjach Open Source jak i komercyjnych. Pełnią ważną rolę w stymulowaniu rozwoju technologicznego i wzrostu konkurencyjności. Otwarte standardy cechuje m.in.:

- zwiększenie interoperacyjności systemów,
- możliwość integracji danych z wielu źródeł,
- oddzielenie sposobu przechowywania danych od formy ich udostępniania,
- niskie koszty implementacji,
- uniezależnienie dostępu do danych od posiadanego oprogramowania (tzw. *vendor lock-in*).

W środowisku GIS instytucją odpowiedzialną za tworzenie otwartych standardów jest *Open Geospatial Consortium (OGC)*. Jest to międzynarodowa organizacja typu non-profit zrzeszająca firmy, instytucje rządowe oraz jednostki naukowe, a jej celem jest rozwój i implementacja otwartych standardów dla danych i usług przestrzennych.

Otwarte standardy są często opierane o istniejące wcześniej rozwiązania np. format *Geopackage* wykorzystuje bazę danych *SQLite*. Istnieją również otwarte standardy, które nie są przygotowane przez *OGC*, jednak ze względu na wolną licencję, łatwą implementację i popularność stają się de facto ogólnie przyjęte. Przykładami są m.in. format *GeoJSON*, który w ciągu kilku ostatnich lat stał się jednym z najczęściej stosowanych formatów do przesyłania wektorowych danych przestrzennych oraz usługa *VectorTiles* stworzonej przez firmę *Mapbox*.

Do przesyłania danych przestrzennych za pośrednictwem Internetu istnieje kilka standardów. Wśród usług przeglądania można wymienić m.in.:

- **Web Map Service (WMS)** - format *OGC* do udostępniania map w postaci dynamicznie generowanych plików,
- **Web Map Tile Service (WMTS)** - standard *OGC* udostępniania danych w formie predefiniowanych rastrowych kafli,
- **Slippy Maps, Tiles** - dane w formie predefiniowanych rastrowych kafli, wykorzystywane m.in. obsługi podkładów *OpenStreetMap*.

Usługi pobierania to m.in.:

- **Web Feature Service (WFS)** - standard *OGC* oparty na XML, pozwala pobierać, identyfikować i edytować dane wektorowe,
- **Web Coverage Service (WCS)** - standard *OGC* oparty na XML do pobierania danych, które zmieniają się w czasie lub przestrzeni,
- **GeoJSON** - format oparty o JSON do przesyłania danych wektorowych,
- **VectorTiles** - format pozwalający udostępniać dane wektorowe w formie kafli co przyspiesza ich wyświetlanie.

Otwarte standardy są implementowane w różnych projektach Open Source np:

- **PostGIS** wspiera wszystkie obiekty i funkcje zdefiniowane w specyfikacji *OGC Simple Feature Access*,
- **Geoserver** pozwala publikować dane w formie usług m.in. *WMS, WMTS, WFS*,
- Projekt **QGIS** obsługuje otwarte usługi zarówno jako aplikacja kliencka (*QGIS Desktop*) jak i serwerowa (*QGIS Server*),
- **OpenLayers** pozwala wyświetlać na stronach WWW dane udostępniane w otwartych standardach m.in. *WMS, WMTS, WFS, GeoJSON*.

Otwarte standardy są również zaimplementowane w oprogramowaniu komercyjnym m.in. firm *ESRI, Oracle, Geomedia* czy *Intergraph*, dzięki czemu możliwe jest tworzenie systemów w różnych konfiguracjach - oparte tylko na Open Source, aplikacjach komercyjnych lub rozwiązania hybrydowe.

Dane przestrzenne w systemach webgis

W systemach webgis można wyróżnić dwa kryteria w sposobie dostępu do danych:

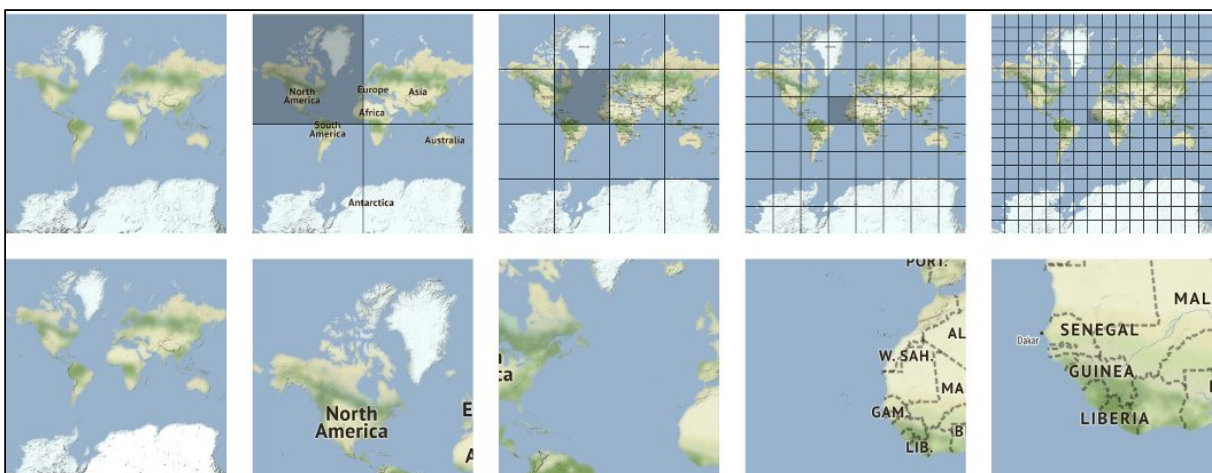
- Rodzaj danych: raster lub wektor,
- Sposób pobierania danych: proste i w formie kafli (*tiles*).

Rastry są to po prostu pliki graficzne posiadające referencję, dzięki której możliwe jest ich wyświetlenie na mapie w określonej lokalizacji. Najczęściej stosuje się dwa formaty graficzne JPEG i PNG, ale możliwe jest stosowanie w zasadzie każdego typu obsługiwanego przez przeglądarkę (np. TIFF, GIF). Format JPEG posiada stratną kompresję, z tego powodu najlepiej sprawdza się do danych ze zobrażeń np. ortofotomapa. PNG obsługuje kompresję bezstratną więc nadaje się do obsługi danych gdzie wartości pikseli mają znaczenie (numeryczne modele terenu) oraz wspiera przezroczystość.

W kwestii pobierania proste dane są pobierane w całości lub wg określonych kryteriów (np. widoczny fragment mapy). Istnieje jednak możliwość generowania po stronie serwera tzw. kafli z danych źródłowych. Są to prostokątne (najczęściej kwadratowe) fragmenty danych, które są przechowywane w formie osobnych plików lub rekordów w bazie. Kafle mogą być określone dla każdego układu współrzędnych płaskich prostokątnych, w Internecie najpopularniejszym układem jest *Web Mercator* (EPSG 3857) obejmujący cały świat.

Cały świat podzielony jest na kwadraty o jednakowych rozmiarach i poziomy skalowe. Zerowy poziom skalowy obejmuje cały świat i jest to pojedynczy kafelek. Pierwszy poziom dzieli świat na 4 kafelki - poprzez równy podział wyższego kafełka poziomu. Podobnie poziom drugi to 16 kafełków - czyli każdy z czterech kafli poziomu 1 jest podzielony na kolejne cztery itd. Liczba kafli na każdym poziomie rośnie wykładniczo i np. na poziomie 20 jest to ponad 1 bilion kafli. Oczywiście dotyczy to całego świata, jeśli mamy dane dla mniejszego obszaru to nie ma potrzeby ich tworzenia dla innych miejsc. Również w przypadku układów, które obejmują jedynie pewien fragment świata (np. PUWG 1992 dla terytorium Polski) liczba kafli do pokrycia całego obszaru jest znacznie mniejsza.

Kafle są zazwyczaj wcześniej wygenerowane i przechowywane na serwerze i mogą być zwracane jak zwykle pliki statyczne. Przykład podziału świata w regularnej siatce dla poziomów od 0 do 4:



Źródło: <https://medium.com/planet-stories/a-gentle-introduction-to-gdal-part-3-geodesy-local-map-projections-794c6ff675ca>

Każdy taki wycinek ma zdefiniowaną lokalizację za pomocą trzech współrzędnych XYZ. Współrzędne XY określają kolumnę i wiersz, natomiast Z to poziom skalowy. Przy obsłudze

danych w formie kafli aplikacja kliencka sama decyduje, które kafle ma pobrać dla widocznego obszaru mapy i wysyła osobne zapytania dla każdego kafła niezależnie. Powoduje to szybsze pobieranie nowych informacji (np. po zmianie widoku).



Źródło: https://wiki.openstreetmap.org/wiki/Slippy_map_tilenames

1. Rastry

Do serwowania prostych rastrów lub kompozycji mapowych najczęściej wykorzystuje się usługę *WMS*. Jest ona jednym z najstarszych i najbardziej rozpowszechnionych sposobów przekazywania informacji geograficznej poprzez Internet. Pozwala pobrać fragment mapy w formie obrazu zawierającego georeferencję, czyli informacje o lokalizacji w przestrzeni w danym układzie współrzędnych. Aplikacja kliencka w żądaniu określa m.in. zasięg, układ współrzędnych, style i rozmiar obrazu. W odpowiedzi otrzymuje pojedynczy plik reprezentujący dany obszar.

Główne zalety usługi WMS:

- + umożliwia wyświetlenie na mapie (zwracany plik graficznym) wielu warstw,
- + wspierana jest przezroczystość w przypadku korzystania z formatu PNG,
- + możliwość wykorzystania różnych stylizacji do tego samego zestawu danych,
- + łatwa możliwość dodawania nowych układów współrzędnych,
- + można wyświetlać dane rastrowe i wektorowe renderowane na obrazie,

Wady usługi WMS:

- duże obciążenie dla serwera - mapa zazwyczaj jest generowana po jego stronie i obciąża cały system,
- brak możliwości sterowania wyglądem po stronie aplikacji klienckiej, np. wyłączenie/zmiana etykiet lub zmiana koloru linii,
- utrudnione buforowanie danych w celu przyspieszenia generowania widoku mapy.

2. Kafle rastrowe

Do obsługi kafli rastrowych stosuje się otwarty standard *OGC WMTS* oraz usługi *TMS* i *XYZ*, które różnią się implementacją i sposobem przechowywania danych, ale od strony aplikacji klienckiej ich działanie jest zbliżone.

Zalety:

- + kafle są serwowane jako pliki statyczne, w związku z czym serwer jest minimalnie obciążony przez zapytania,
- + wspierana jest przezroczystość w przypadku korzystania z formatu PNG,
- + aplikacja kliencka może równocześnie pobierać kilka kafli co przyspiesza wyświetlanie całości mapy,
- + można wyświetlać dane rastrowe i wektorowe renderowane na obrazie.

Wady:

- każda zmiana w danych, stylach lub dodanie układu współrzędnych wymaga aktualizacji lub stworzenia nowego zestawu plików,
- dane mogą zajmować dużo miejsca na dysku twardym,
- najlepiej wyglądają w predefiniowanych poziomach zbliżenia, w innych skalach mogą być nieostre.

3. Wektory

Proste dane wektorowe są pobierane w całości przez aplikację kliencką i wyświetlane na mapie. Najczęściej do wymiany informacji wektorowych stosuje się format *GeoJSON* i *GML*, rzadziej *KML/KMZ*. Istnieje otwarty standard *OGC WFS*, który służy do pobierania, edycji i tworzenia danych wektorowych. Domyślnie wykorzystuje on format *GML*.

Zalety:

- + można stylizować dane dynamicznie po stronie aplikacji klienckiej,
- + możliwość edycji danych,
- + dane pobierane są za jednym razem, przez co serwer jest mniej obciążony,

- + łatwe buforowanie danych, zarówno po stronie serwera jak i aplikacji klienckiej.

Wady:

- renderowanie danych odbywa się po stronie aplikacji klienckiej, więc przy słabszych podzespołach może działać nieco wolniej niż dane rastrowe,
- przy dużych i skomplikowanych zestawach danych mogą występować w aplikacji klienckiej opóźnienia w wyświetlaniu i spadki wydajności działania (chwilowe zawieszanie).

4. Kafle wektorowe

Zalety:

- + można stylizować dane dynamicznie po stronie aplikacji klienckiej,
- + zoptymalizowane pod kątem wyświetlania dużych zbiorów danych wektorowych,
- + aplikacja kliencka może równocześnie pobierać kilka kafli co przyspiesza wyświetlenie całości mapy,
- + mały rozmiar danych w porównaniu do kafli rastrowych,
- + szybszy czas generowania w porównaniu do kafli rastrowych, ponieważ nie wymagają renderowania,
- + możliwość generalizacji danych dla poszczególnych poziomów przybliżenia, co zmniejsza rozmiar danych i wydajność ich przetwarzania,
- + łatwa możliwość dodawania nowych układów współrzędnych,
- + łatwe buforowanie danych, zarówno po stronie serwera jak i aplikacji klienckiej.

Wady:

- renderowanie danych odbywa się po stronie aplikacji klienckiej, więc przy słabszych podzespołach może działać nieco wolniej niż dane rastrowe,
- na chwilę obecną brak standardu OGC dla tego typu danych, prace nad specyfikacją trwają,
- niezbyt dobrze nadają się do edycji m.in. ze względu na generalizację danych.

Baza danych przestrzennych PostGIS

PostgreSQL to jeden z najpopularniejszych systemów zarządzania relacyjnymi bazami danych (tzw. *RDBMS*). Projekt jest rozwijany od lat 80-tych pod nazwą Ingres, obecną nazwę uzyskał 1996 r.

PostgreSQL wspiera system rozszerzeń, dzięki któremu możliwe jest m.in. dodawanie nowych funkcji języka SQL i typów danych. W 2005 r. została wydana pierwsza wersja rozszerzenia *PostGIS*, która dodała do tej bazy danych obsługę danych przestrzennych. Opiera się ona na specyfikacji *OpenGIS Simple Features* dla języka SQL, która została opracowana przez organizację *Open Geospatial Consortium* (OGC).

Obecnie *PostgreSQL* i *PostGIS* oferują największe możliwości ze wszystkich otwartych baz danych w obsłudze danych przestrzennych. Z tego względu jest to jedno z najpopularniejszych rozwiązań wykorzystywanych przy tworzeniu systemów webgis.

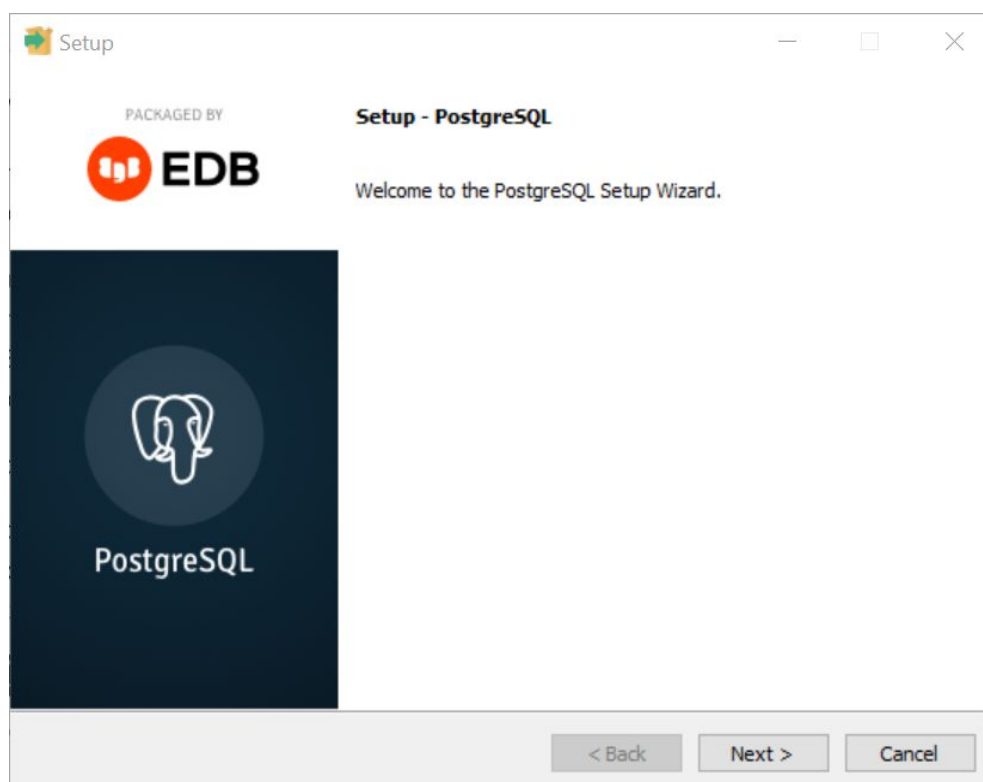
Instalacja w systemie Windows

W systemach operacyjnych z rodziny Windows PostgreSQL wraz z rozszerzeniami należy zainstalować używając instalatorów dostarczanych przez firmę EnterpriseDB. Instalatory te zawierają:

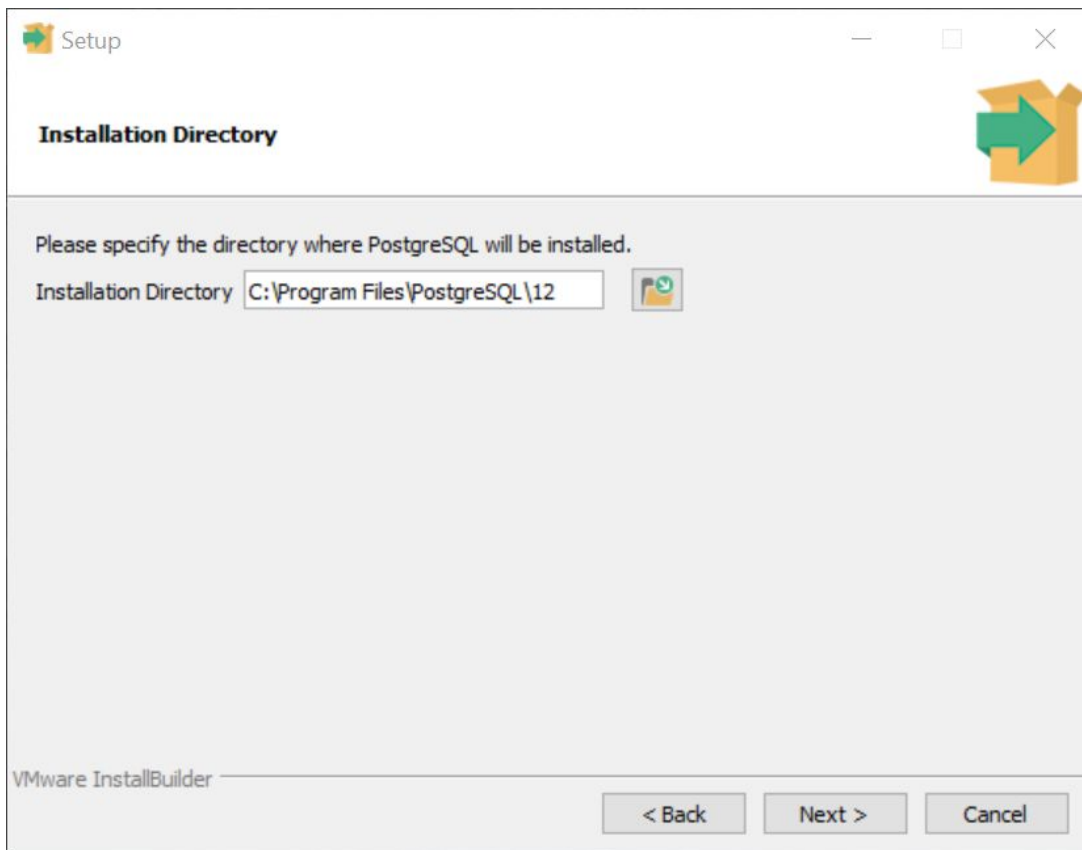
- **serwer bazy danych**
- **pgAdmin** - narzędzie do zarządzania oraz pracy na bazie danych
- **StackBuilder** - manager pakietów pozwalający na pobieranie oraz instalowanie dodatkowych rozszerzeń i sterowników

Instalator można pobrać ze strony <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>, gdzie dostępne są różne wersje bazy na systemy z rodziny Linux, Windows i Mac OS X. Po pobraniu i uruchomieniu instalatora należy potwierdzić uprawnienia do instalacji aplikacji w systemie. Instalacja odbywa się w kreatorze, w którym użytkownik w kolejnych ekranach ustawia opcje konfiguracyjne.

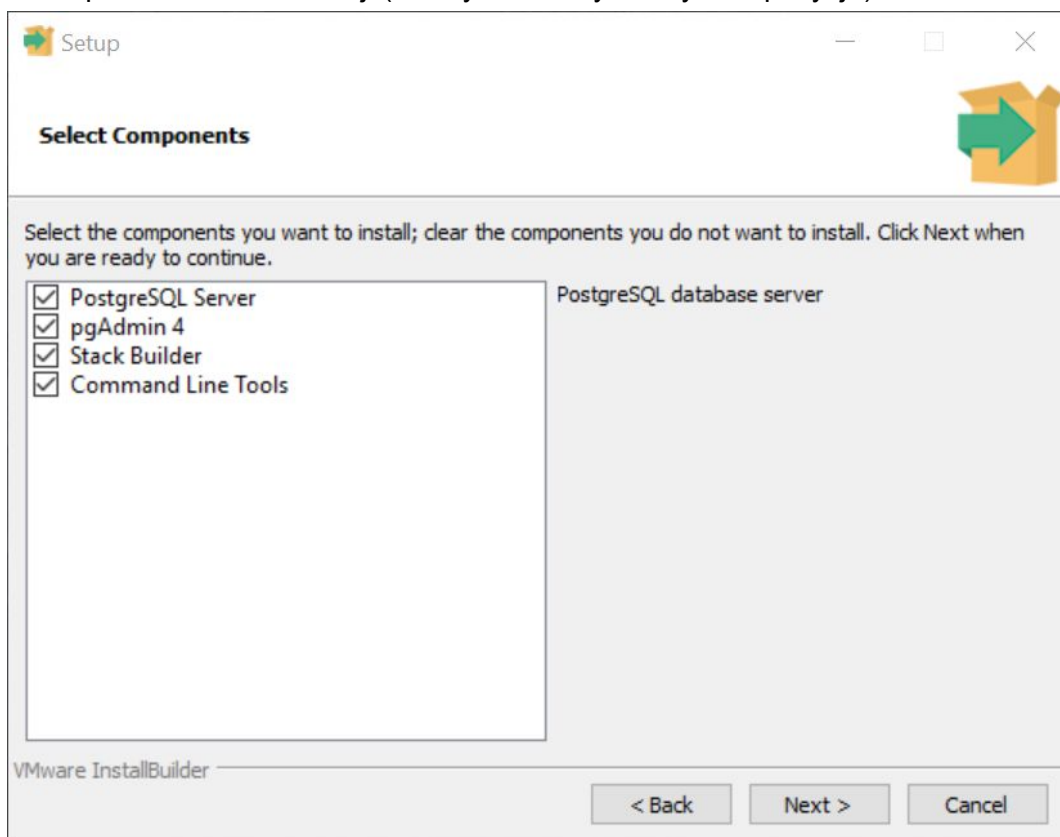
Okno powitalne:



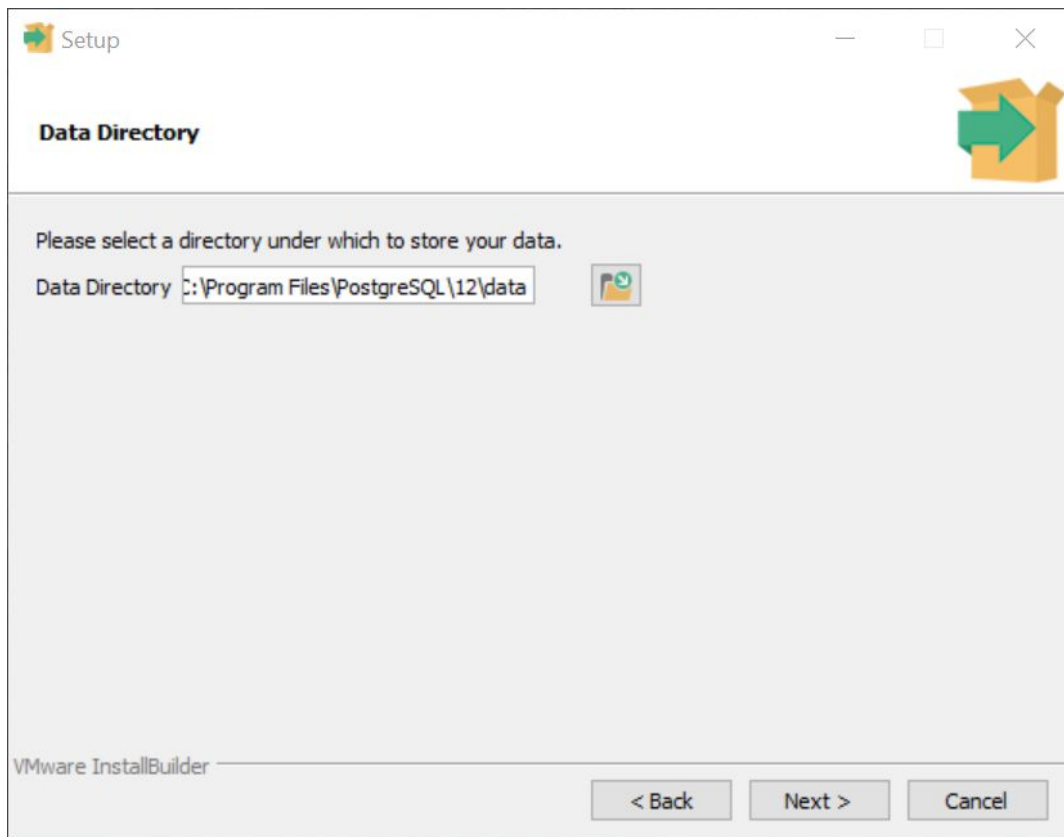
Ustawienie katalogu instalacyjnego:



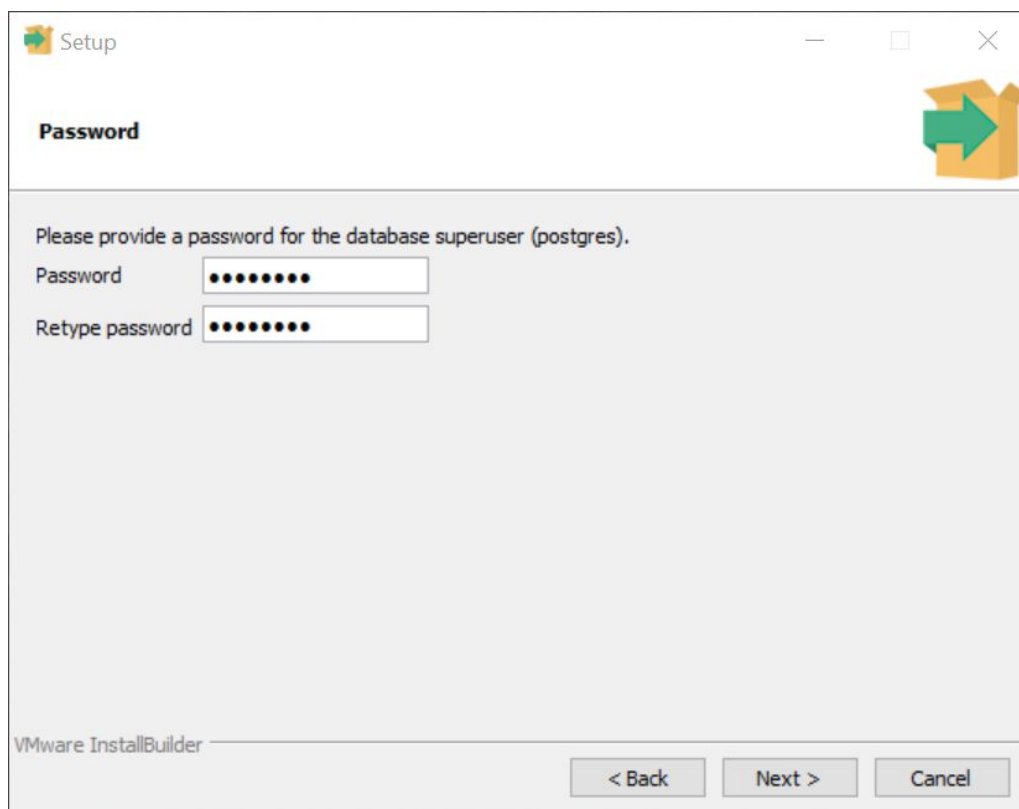
Wybór komponentów do instalacji (należy zaznaczyć wszystkie pozycje):



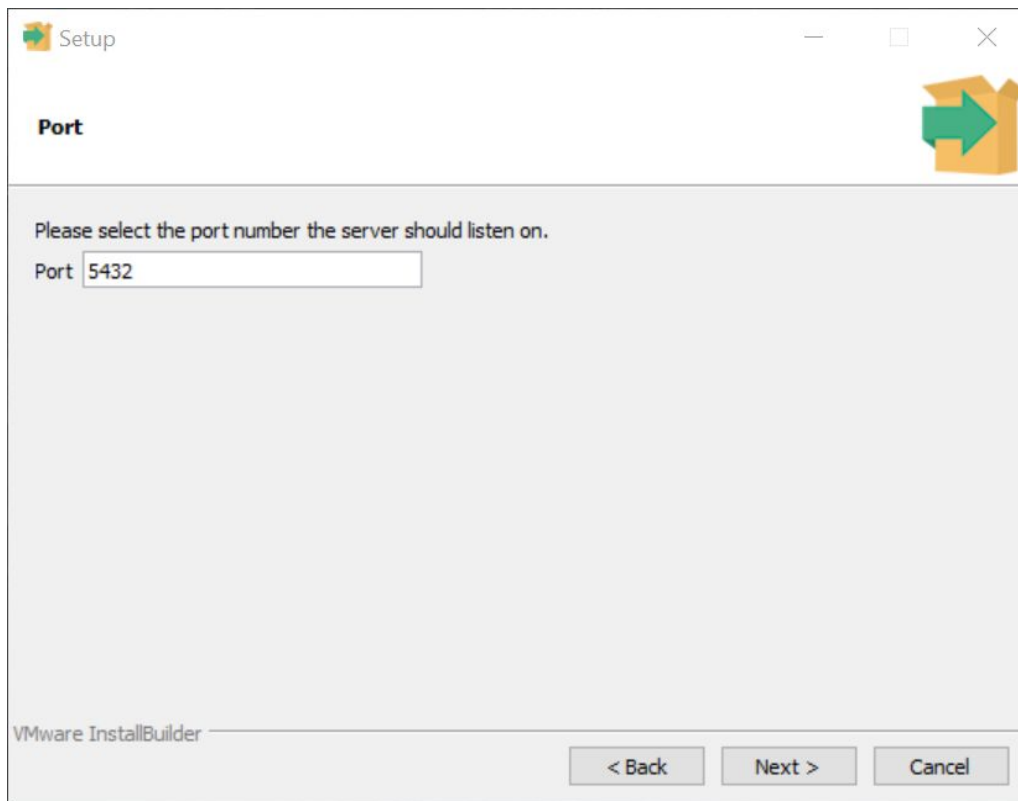
Wybór katalogu, w którym przechowywane będą dane:



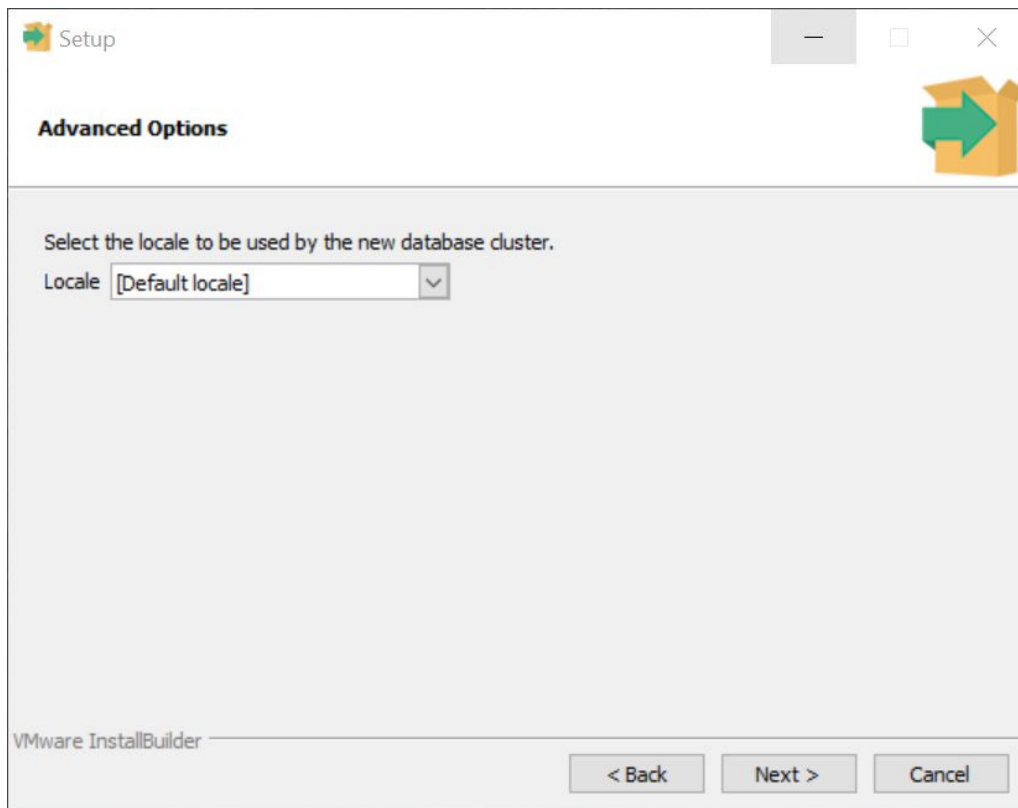
Hasło głównego administratora, można podać dowolne, w celach ćwiczeniowych ustawimy webgis.



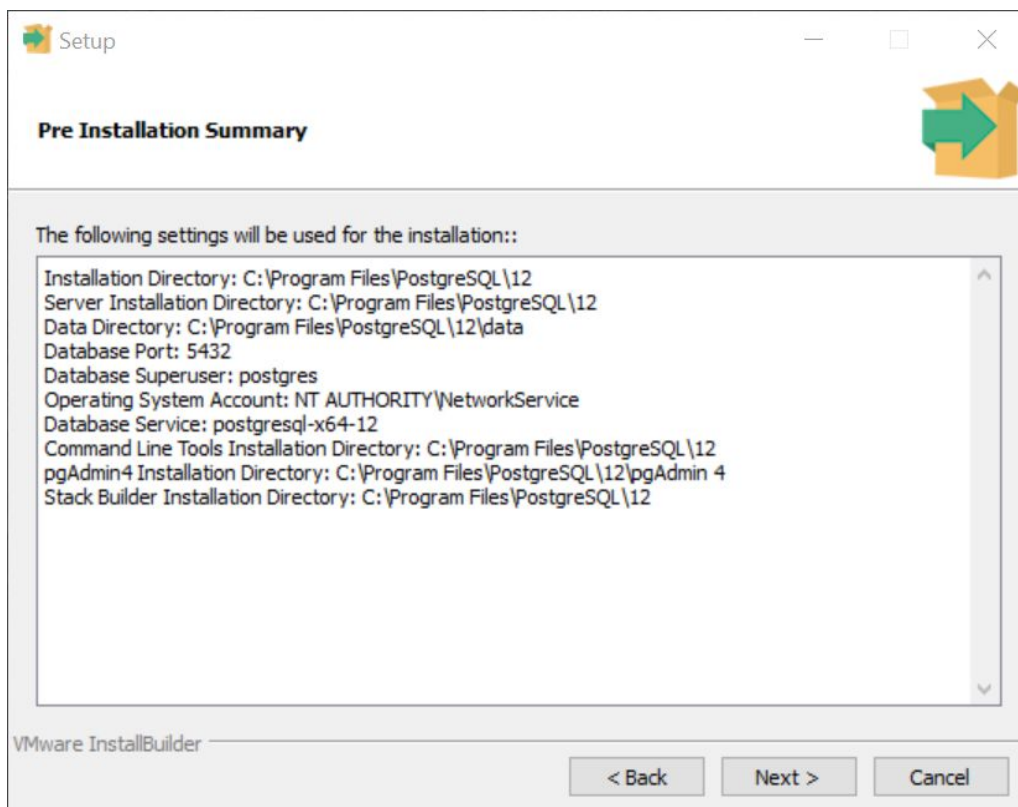
Port, na którym będzie działała baza:



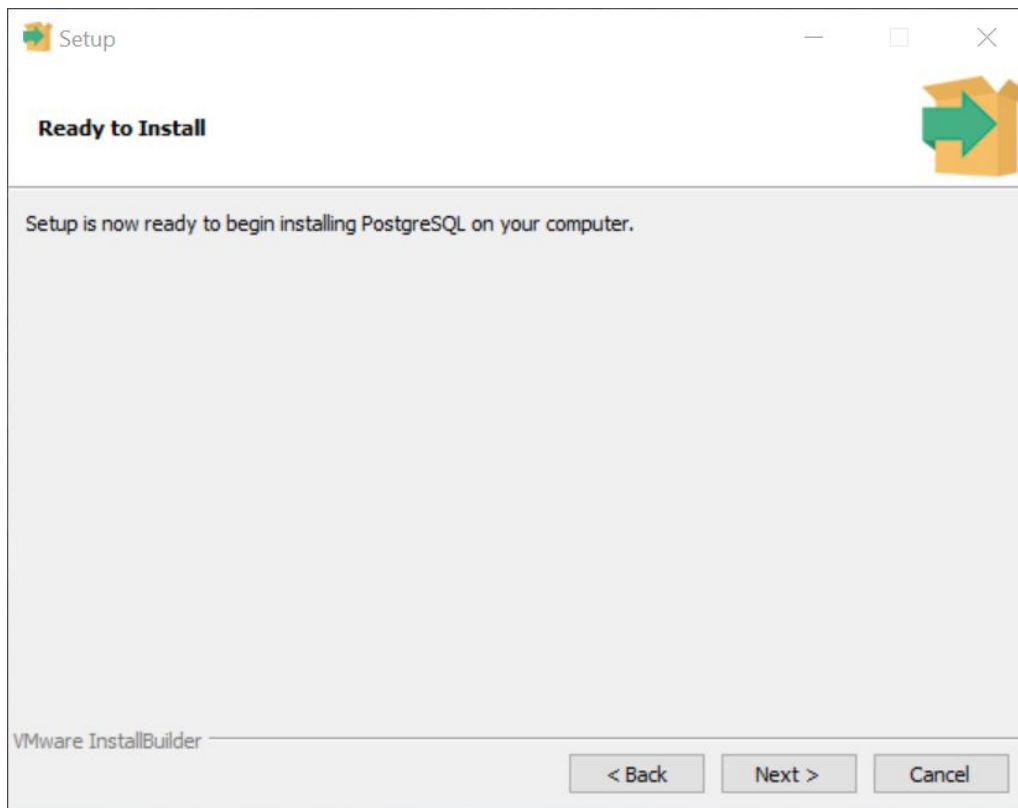
Ustawienia lokalne(pozostawiamy domyślne):



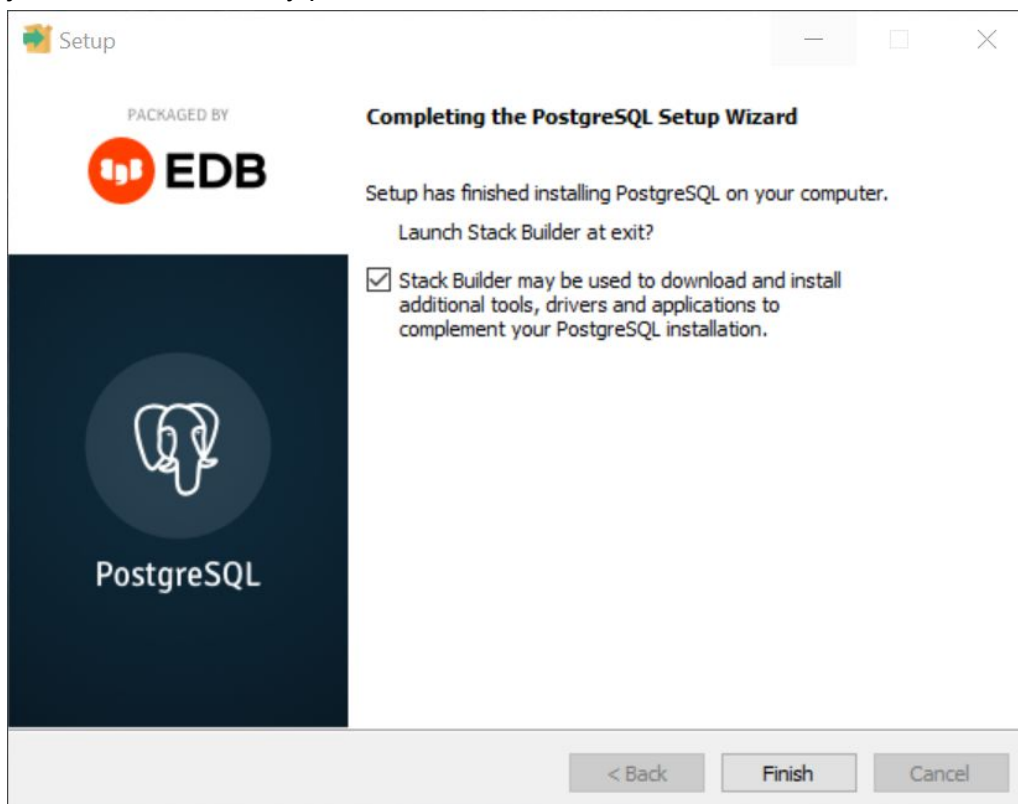
Podsumowanie ustawień:



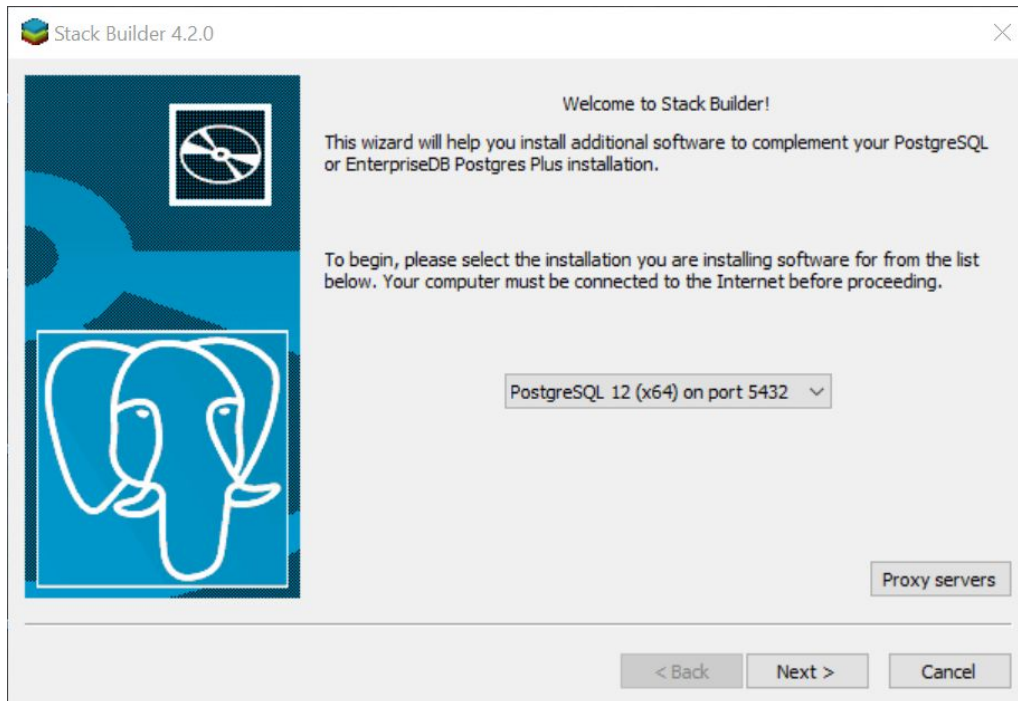
Po kliknięciu przycisku Next rozpocznie się instalacja PostgreSQL:



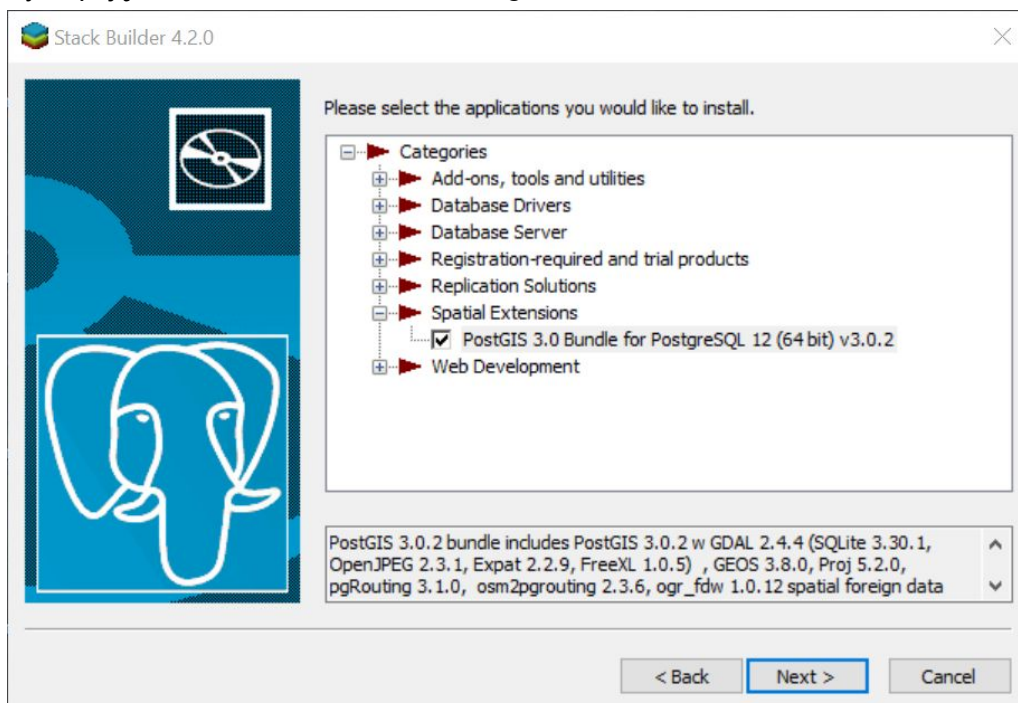
Po zakończeniu instalacji PostgreSQL można zainstalować rozszerzenia za pomocą aplikacji StackBuilder. Należy pozostawić zaznaczenie:



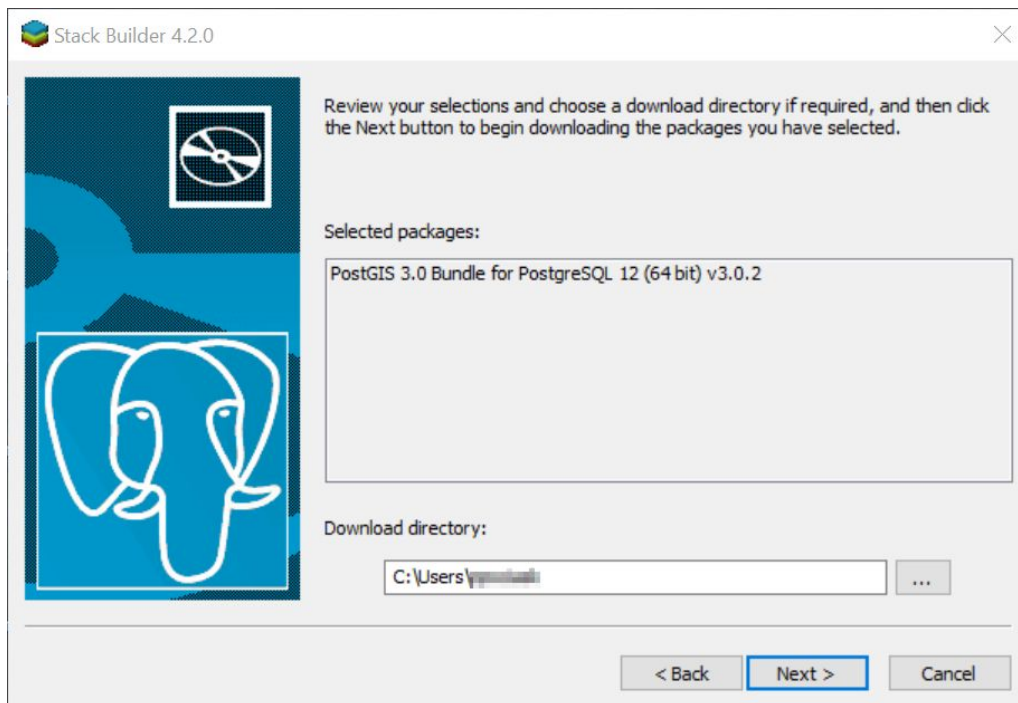
Wskazanie instancji PostgreSQL, dla której odbędzie się instalacja:



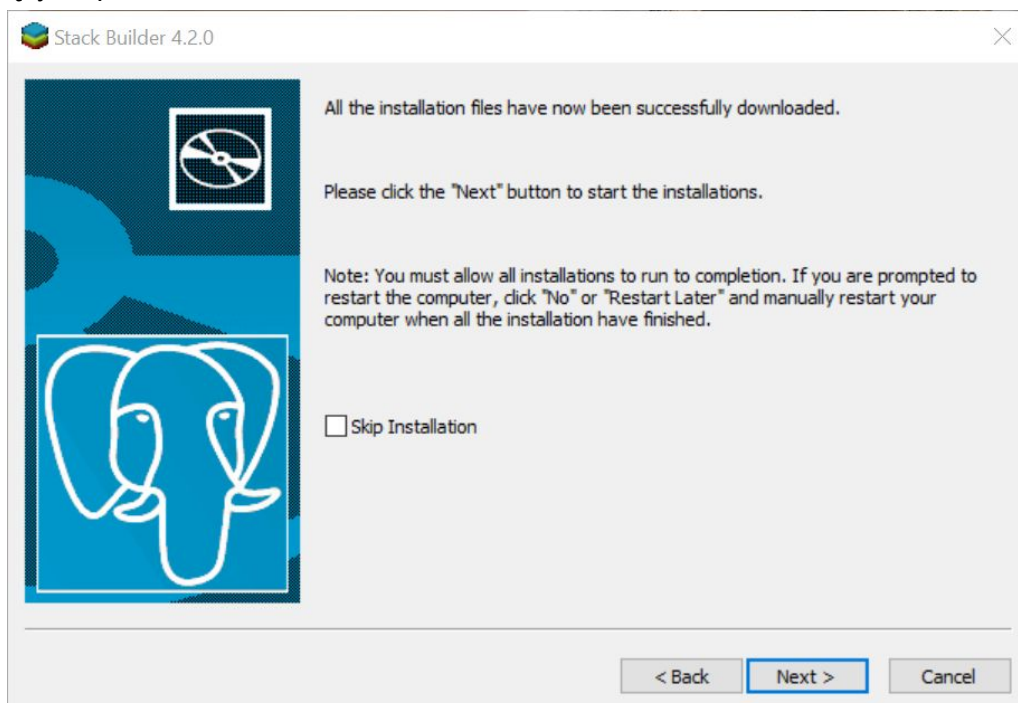
Wybór rozszerzeń. W celu instalacji PostGIS należy rozwinąć opcję *Spatial Extensions* i zaznaczyć opcję *PostGIS 3.0 Bundle for PostgreSQL*.



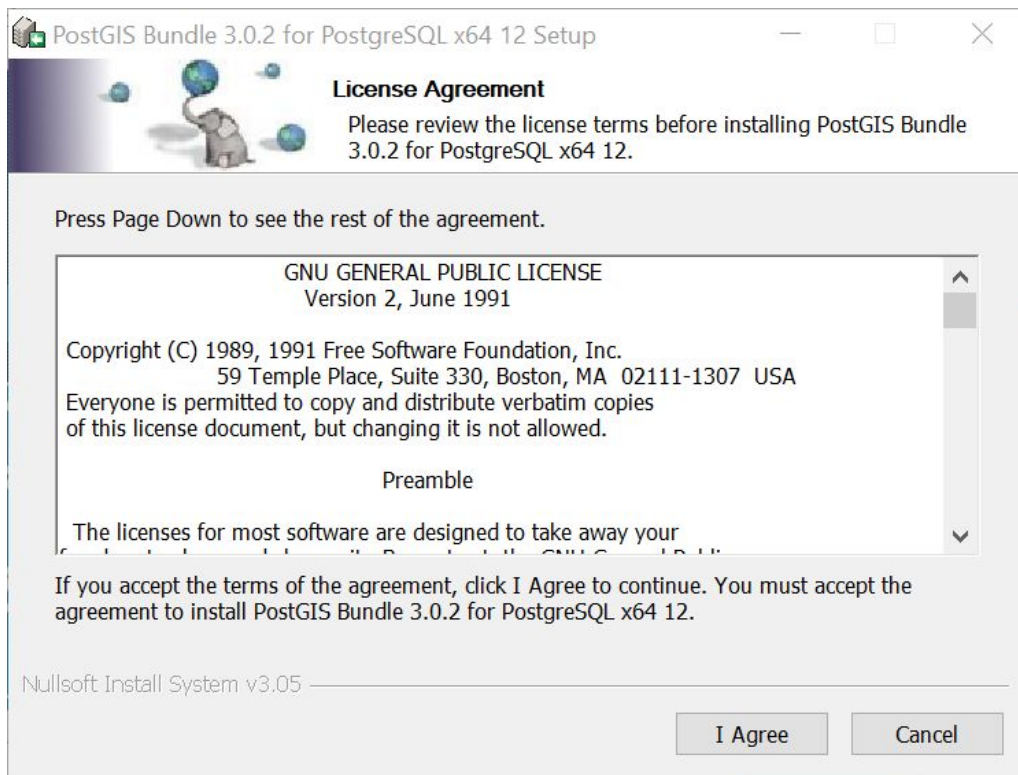
Tymczasowy katalog do pobrania plików:



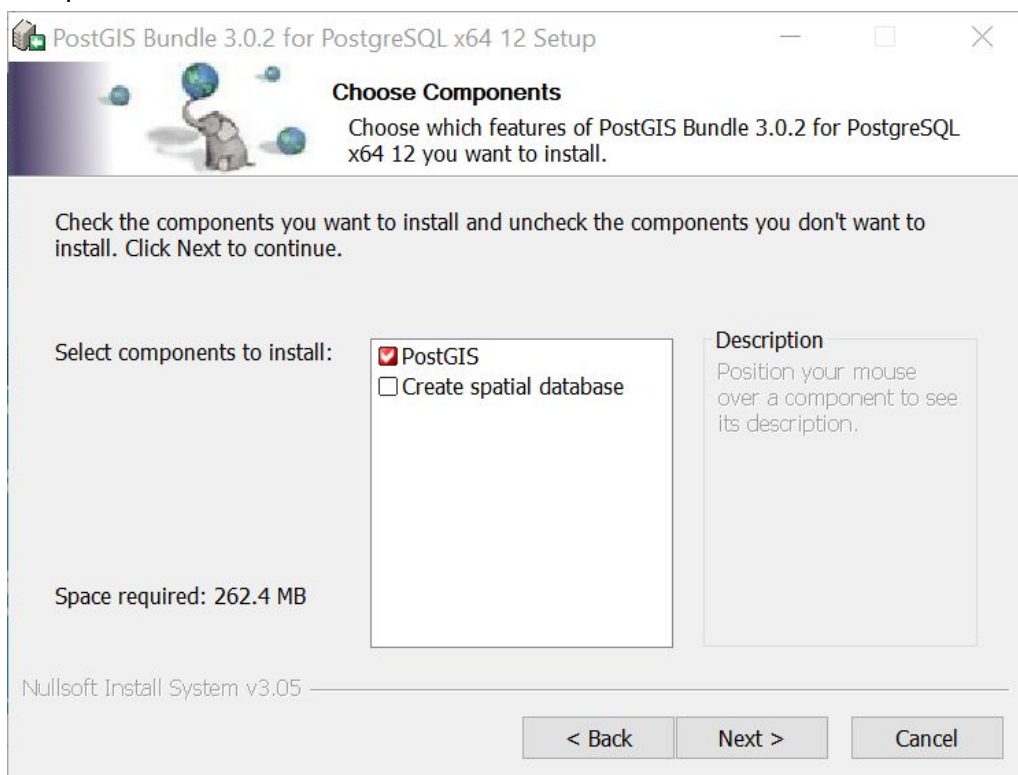
Instalacja rozszerzeń, należy odznaczyć opcję *Skip Installation*, w przeciwnym razie pliki zostaną tylko pobrane:



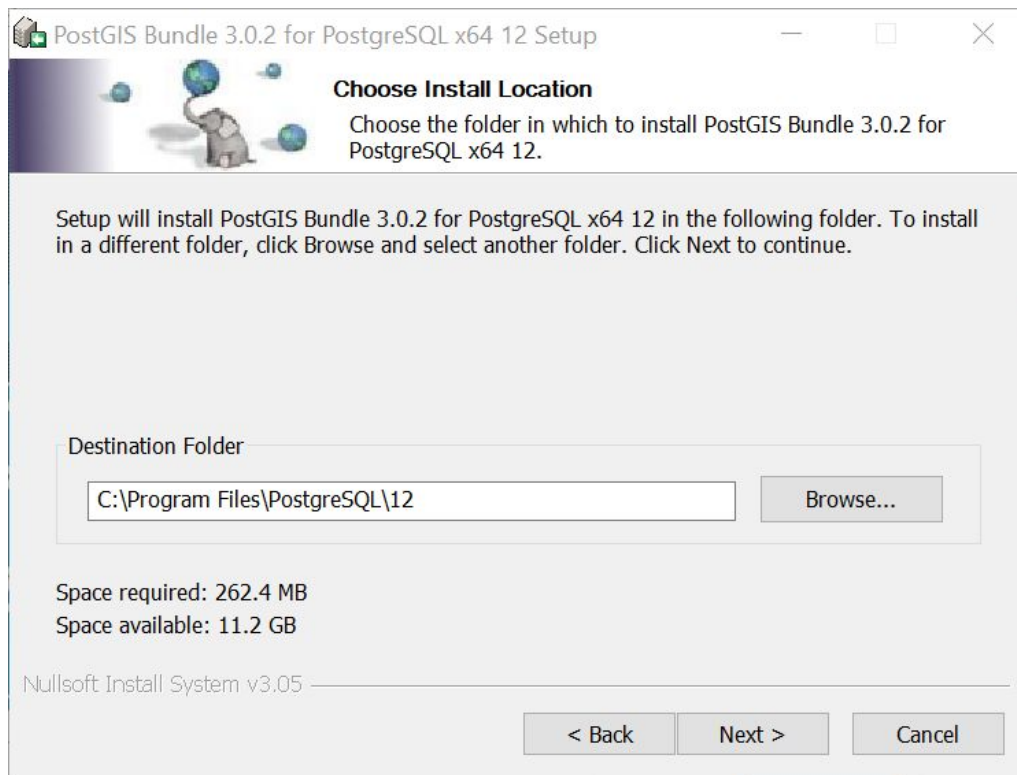
Akceptacja licencji PostGIS:



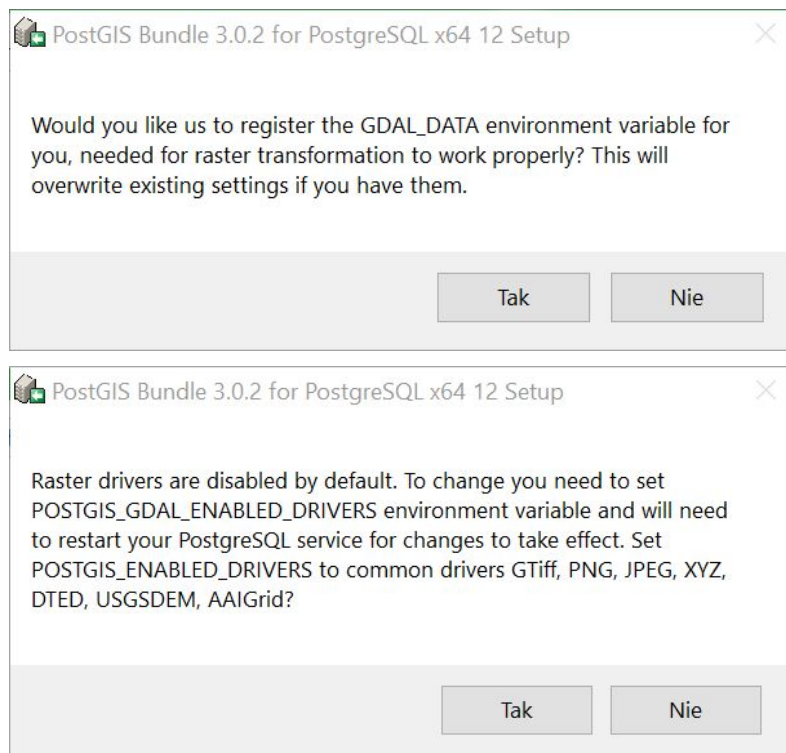
Wybór komponentów do zainstalowania:

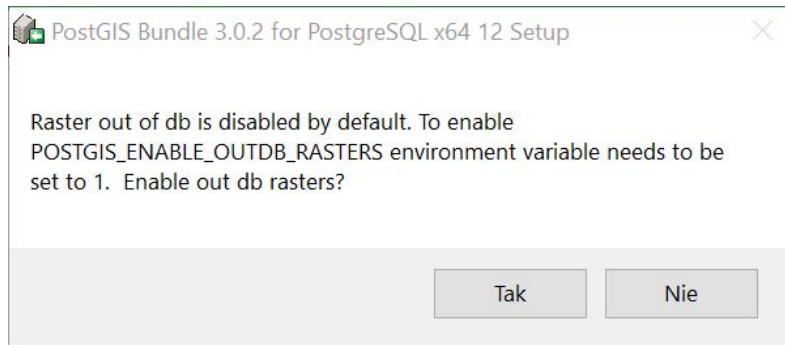


Katalog instalacyjny:

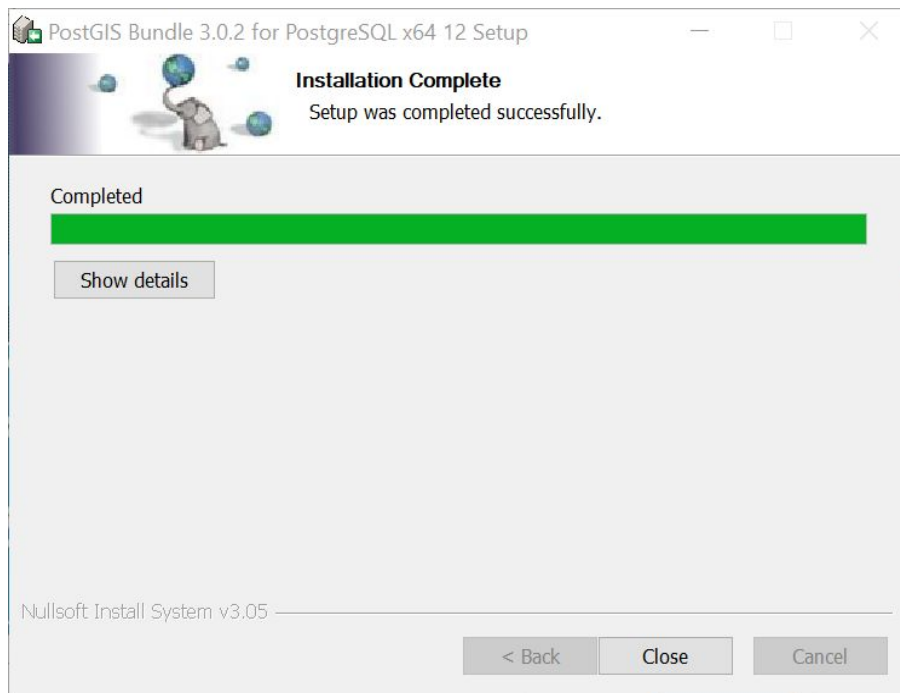


Rejestracja niezbędnych bibliotek w systemie, na wszystkie pytania należy odpowiedzieć twierdząco:

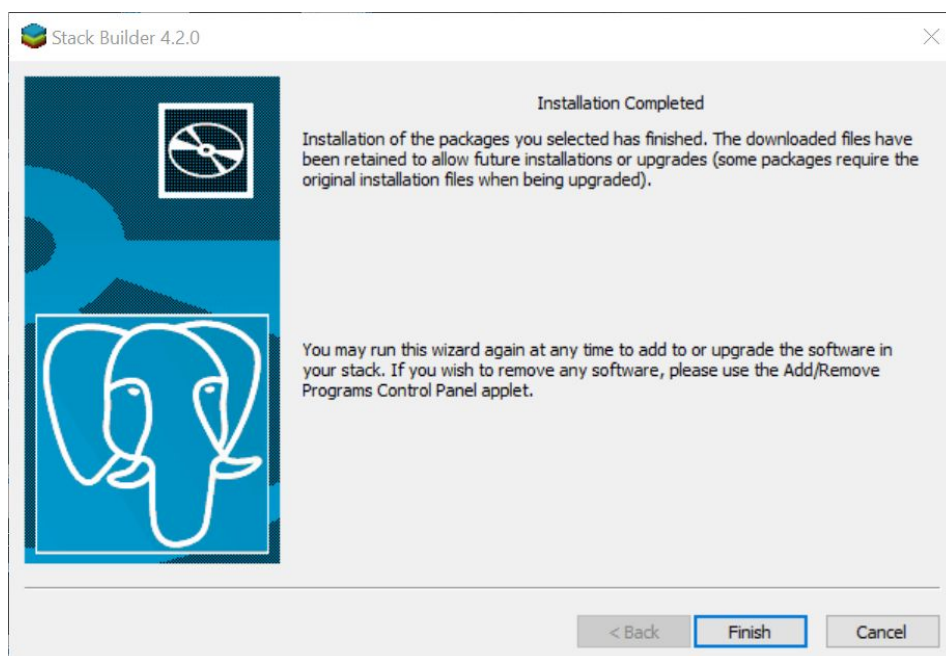




Koniec instalacji PostGIS:



Zamknięcie StackBuilder:

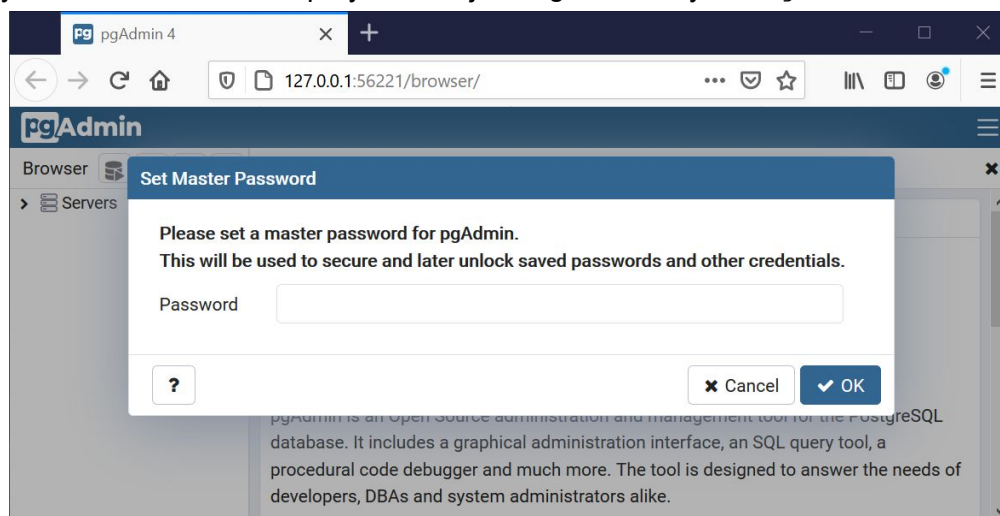


pgAdmin

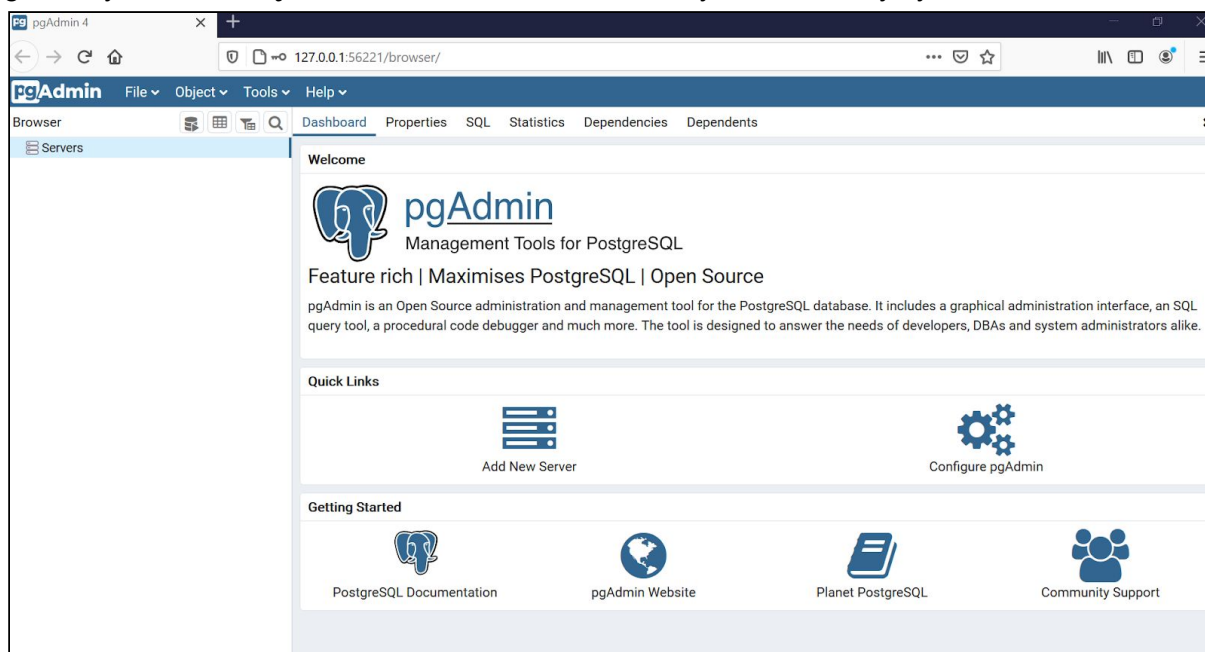
pgAdmin to aplikacja wspomagająca administrowanie i zarządzanie danymi w bazach *PostgreSQL*. Jest to wizualna nakładka na wiersz poleceń do bezpośredniej komunikacji z bazą. Aplikacja pozwala na zarządzanie równocześnie wieloma bazami danych umieszczonymi na wielu serwerach.

pgAdmin dostępny jest w systemowym menu *Start*, po jego włączeniu uruchomi się przeglądarka internetowa z oknem aplikacji.

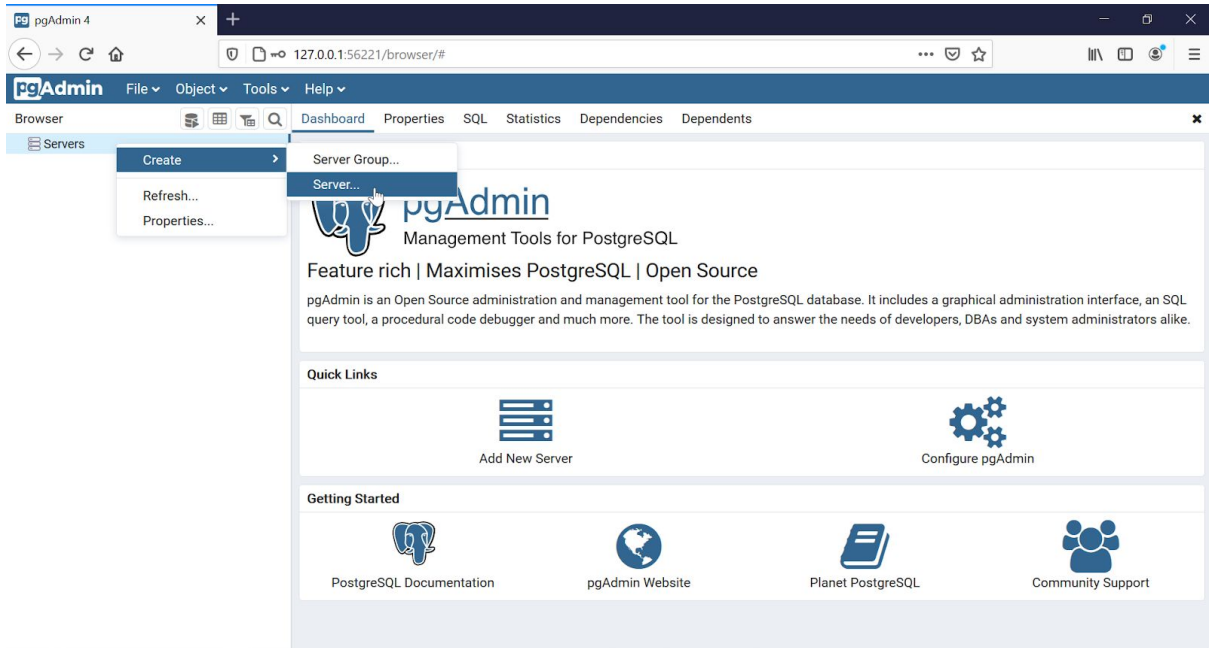
Przy pierwszym uruchomieniu należy podać hasło, które będzie wykorzystywane m.in. do szyfrowania zapisanych danych. Można użyć dowolnego tekstu, do ćwiczeń można wykorzystać to samo hasło co przy instalacji *PostgreSQL* czyli *webgis*.



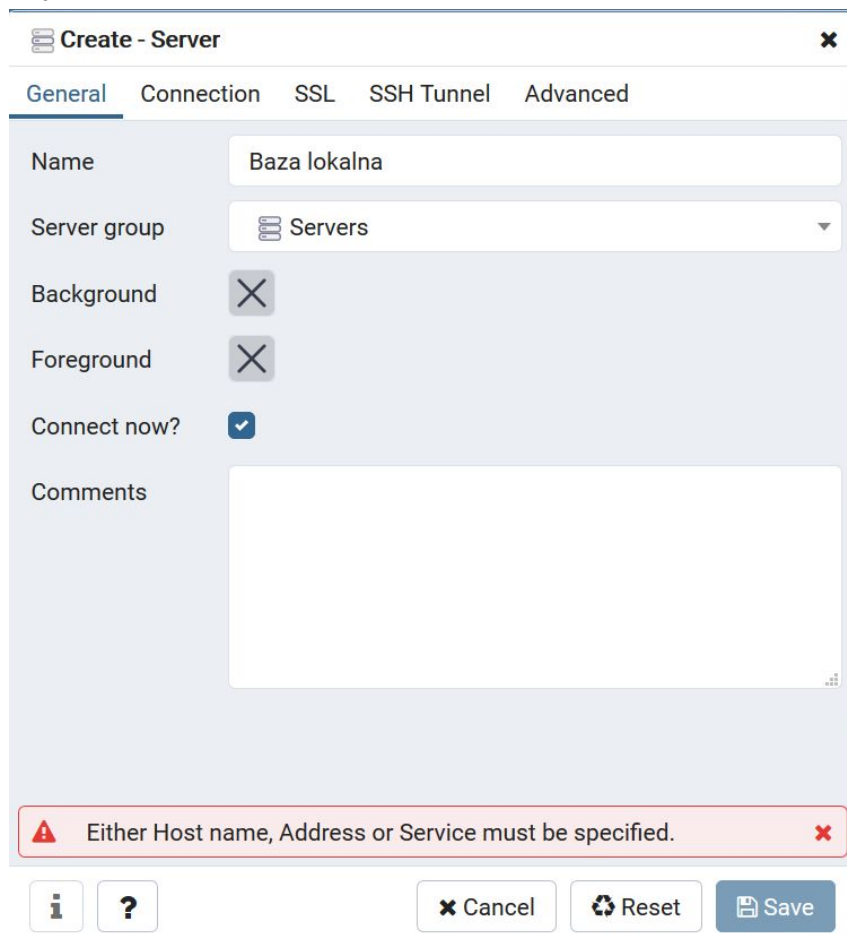
Okno *pgAdmin* podzielone jest na kilka części. Na górze zlokalizowane jest menu, z lewej strony jest przeglądarka dodanych serwerów natomiast z prawej jest widok z zakładkami, gdzie wyświetlane są różne okna m.in. dane tabel, edytor SQL, statystyki i inne.



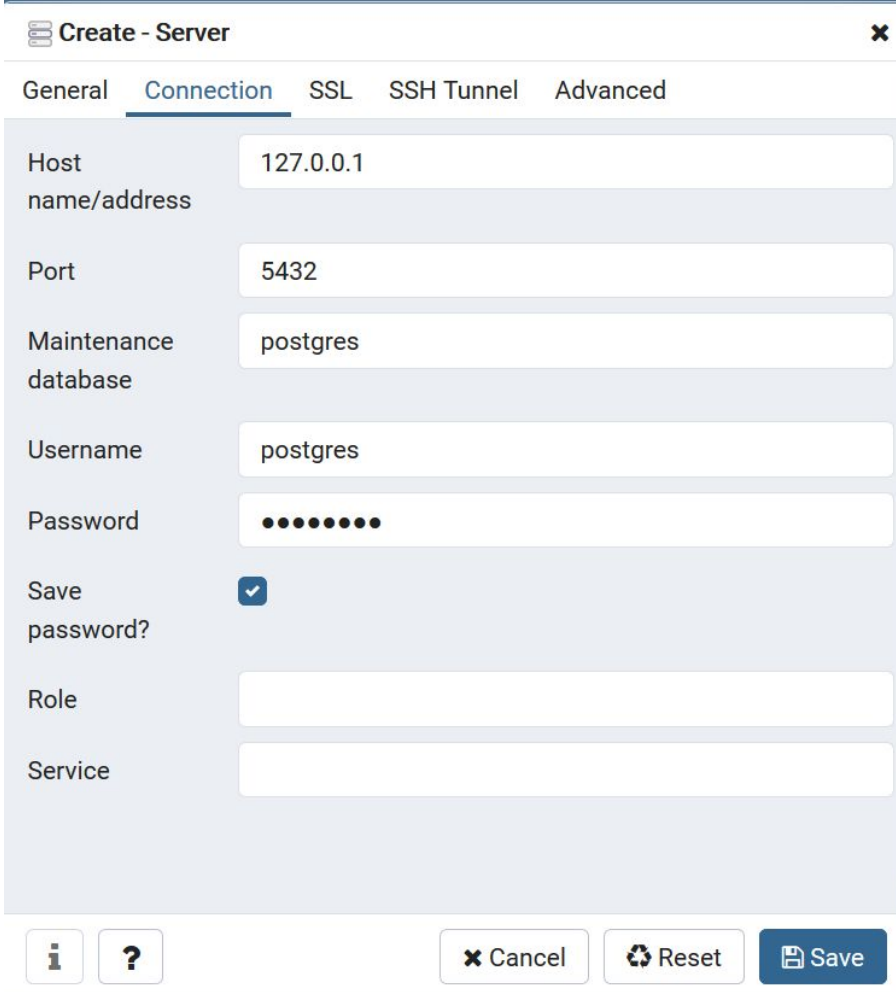
Aby dodać nowy serwer należy kliknąć prawy klawiszem w przeglądarce na istniejącej pozycji *Servers* i wybieramy *Create -> Server*



Wyświetlone zostanie okno konfiguracji połączenia z serwerem. W zakładce *General* podajemy nazwę serwera np. *Baza lokalna*.



Przechodzimy do zakładki *Connection* gdzie podajemy adres serwera 127.0.0.1 (lub localhost) oraz hasło użytkownika postgres które zostało utworzone podczas instalacji - zgodnie z zaleceniami powinno to być hasło webgis.

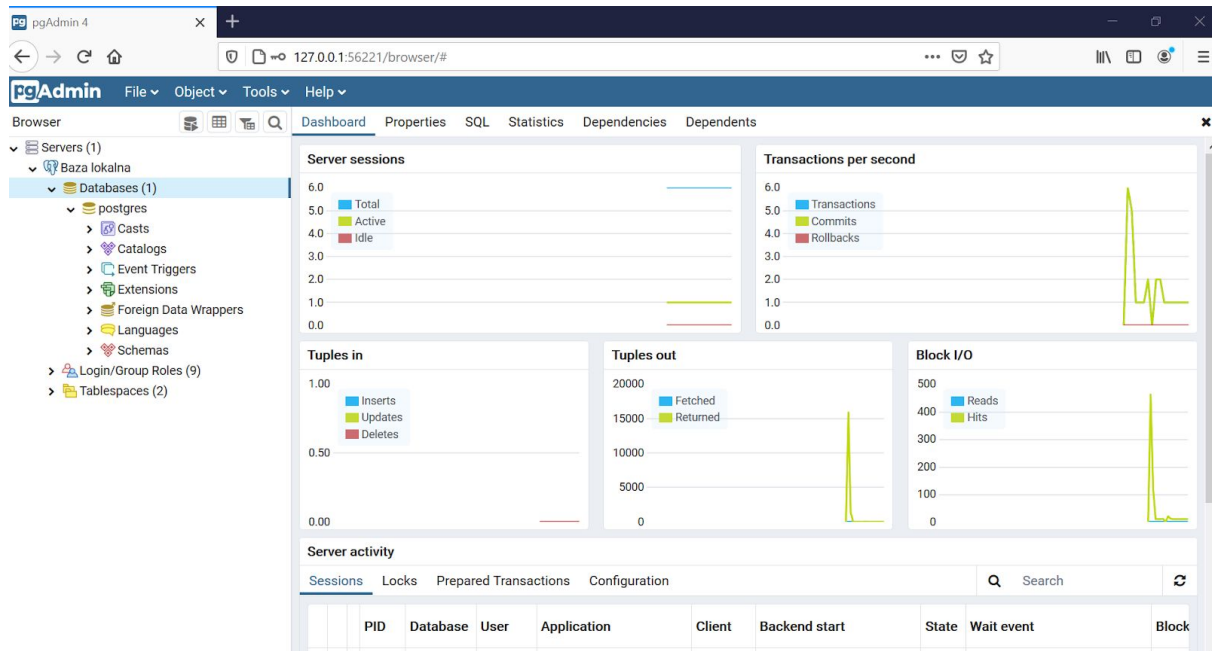


The image shows a 'Create - Server' dialog box with the 'Connection' tab selected. The fields are filled with the following values:

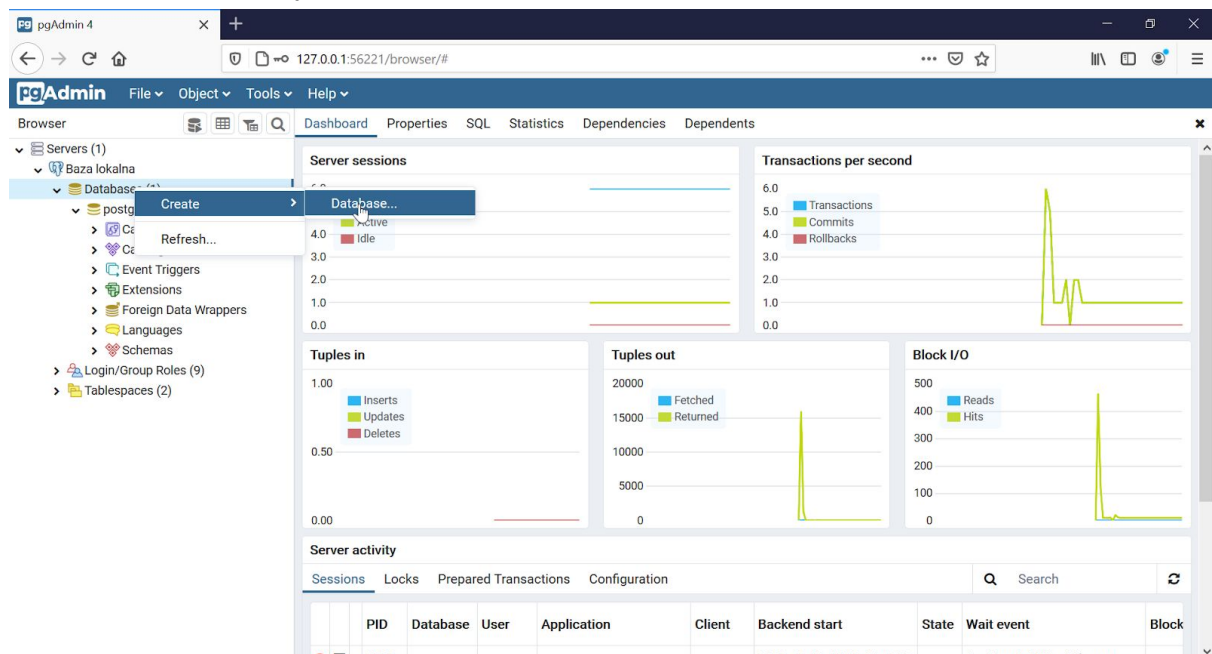
Field	Value
Host name/address	127.0.0.1
Port	5432
Maintenance database	postgres
Username	postgres
Password	••••••••
Save password?	<input checked="" type="checkbox"/>
Role	
Service	

At the bottom of the dialog, there are buttons for 'Cancel', 'Reset', and 'Save', along with information and help icons.

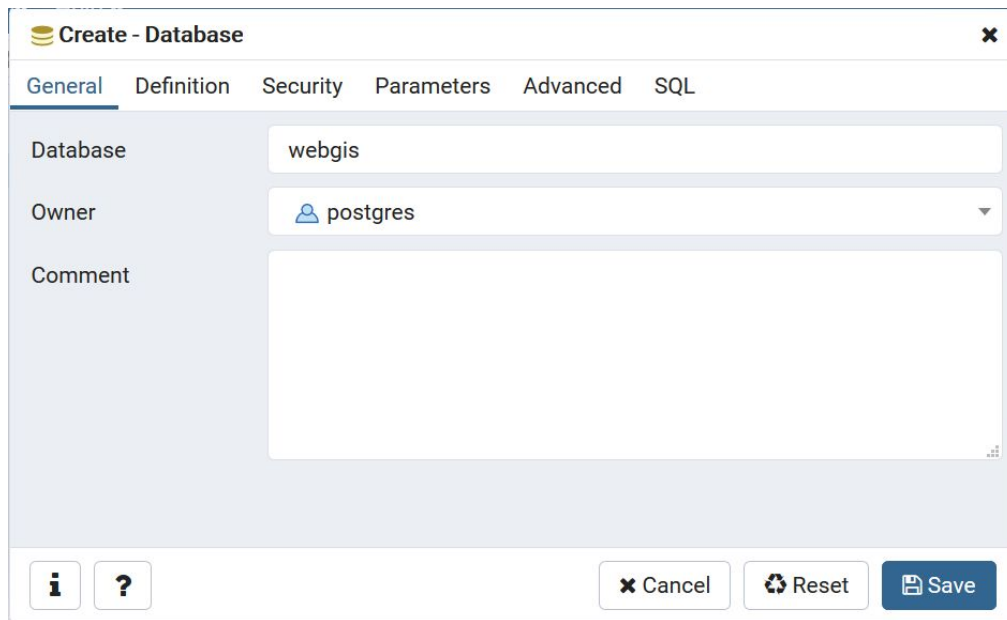
Jeśli podane dane były prawidłowe aplikacja powinna połączyć się do serwera bazy i wyświetlić w drzewie dostępne na tym serwerze bazy danych. Po instalacji dostępna jest tylko jedna baza o nazwie *postgres* i jest to baza serwisowa na której nie powinno się pracować.



Aby utworzyć nową bazę, w której będziemy przechowywać dane, należy kliknąć na *Databases* i z menu wybrać *Create -> Database*



W oknie tworzenia bazy danych w zakładce *General* podajemy jej nazwę - na potrzeby szkolenia sugerujemy nazwę *webgis*, i wybieramy *Save*.



The screenshot shows a 'Create - Database' dialog box with the following fields and options:

- Database:** webgis
- Owner:** postgres
- Comment:** (empty text area)
- Buttons:** Cancel, Reset, Save

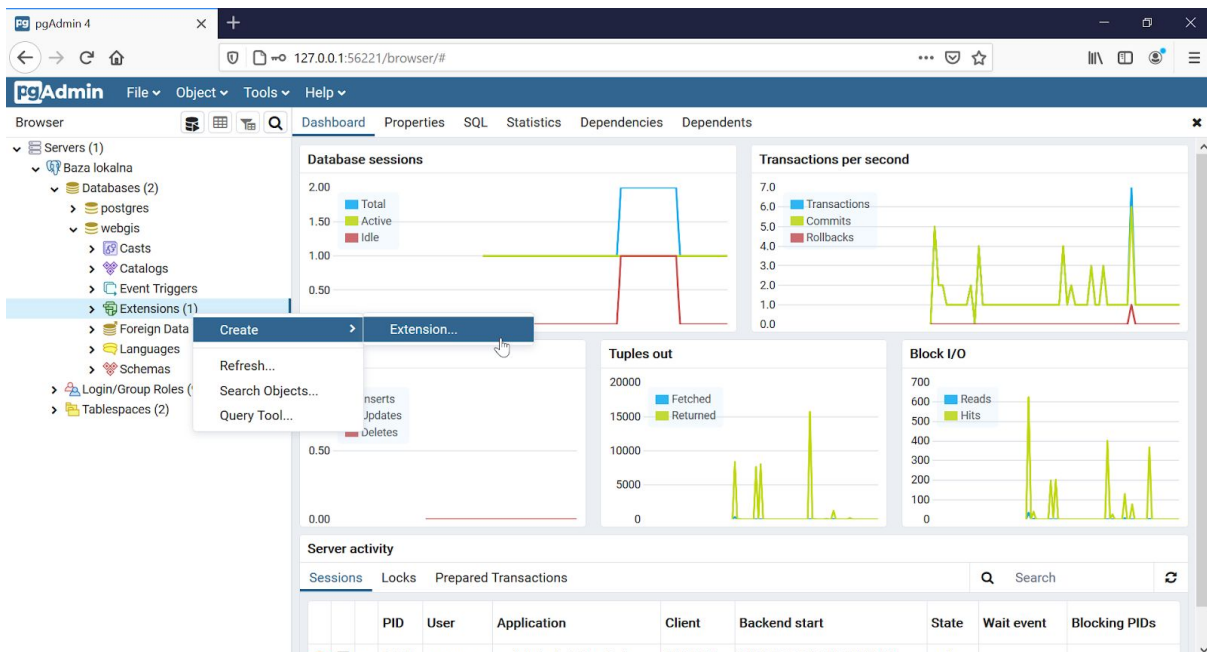
W serwerze powinna zostać utworzona kolejna baza danych o nazwie *webgis*. Domyślnie elementy danej bazy pogrupowane są w kilka pozycji. W naszej perspektywy najważniejsze będą dwa elementy:

- **Extensions** - jest to lista rozszerzeń włączonych dla danej bazy,
- **Schemas** - lista schematów i tabel w bazie.

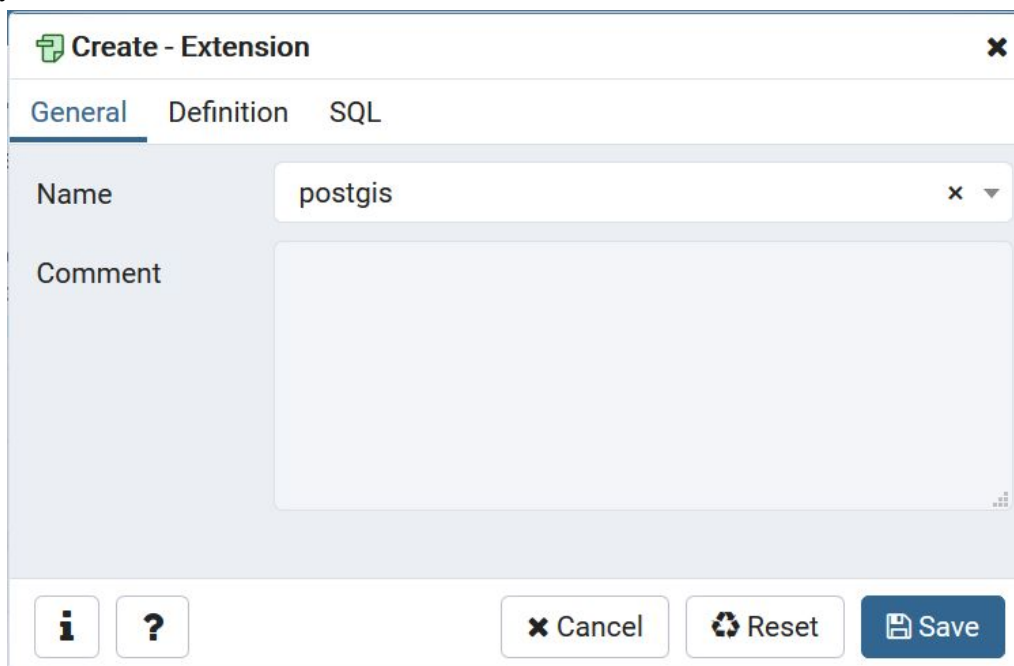
Domyślnie każda nowa baza danych posiada jeden schemat *public*, w którym nie ma żadnych tabel.

PostGIS

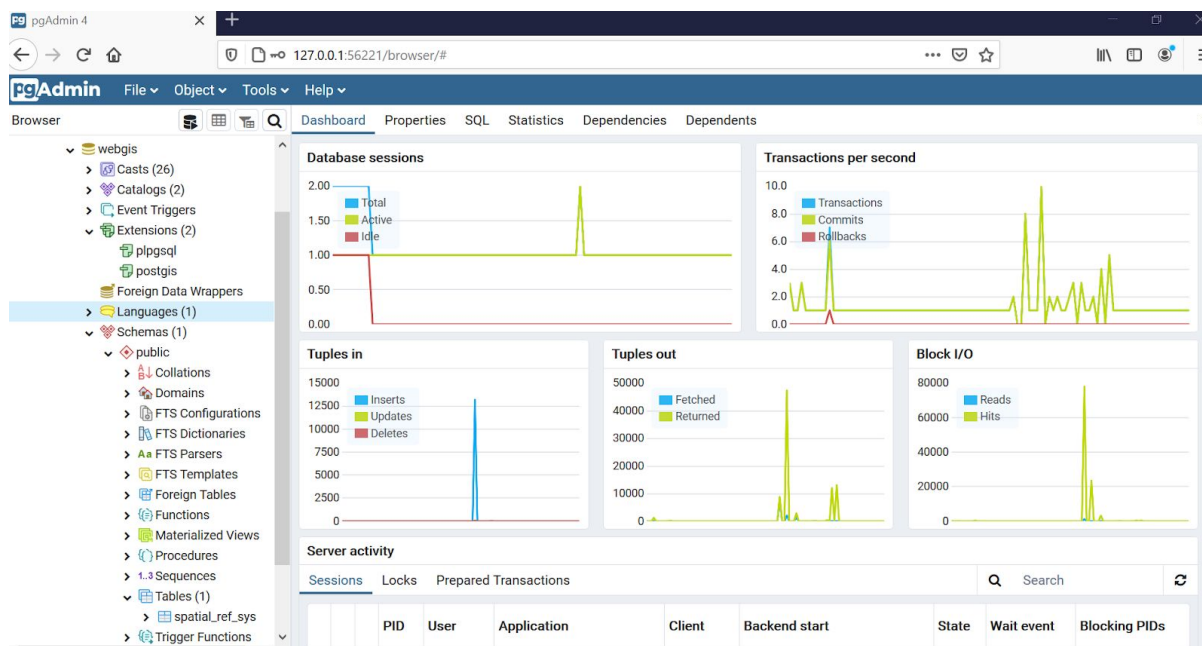
Nasza baza powinna wspierać dane przestrzenne, w związku z tym musimy do niej dodać rozszerzenie *PostGIS*. W tym celu należy kliknąć prawym przyciskiem na *Extensions* i wybrać *Create -> Extension*.



W nowym oknie na zakładce *General* należy znaleźć w liście *Name* wpisać *postgis*. Następnie klikamy *Save*.




Jeśli wszystko poszło bezbłędnie na liście rozszerzeń powinna pojawić się pozycja *postgis*, a w schemacie *public* została utworzona nowa tabela *spatial_ref_sys*. Jest to tabela systemowa dla *PostGIS*, w której znajdują się definicje układów współrzędnych. Nie należy jej modyfikować.



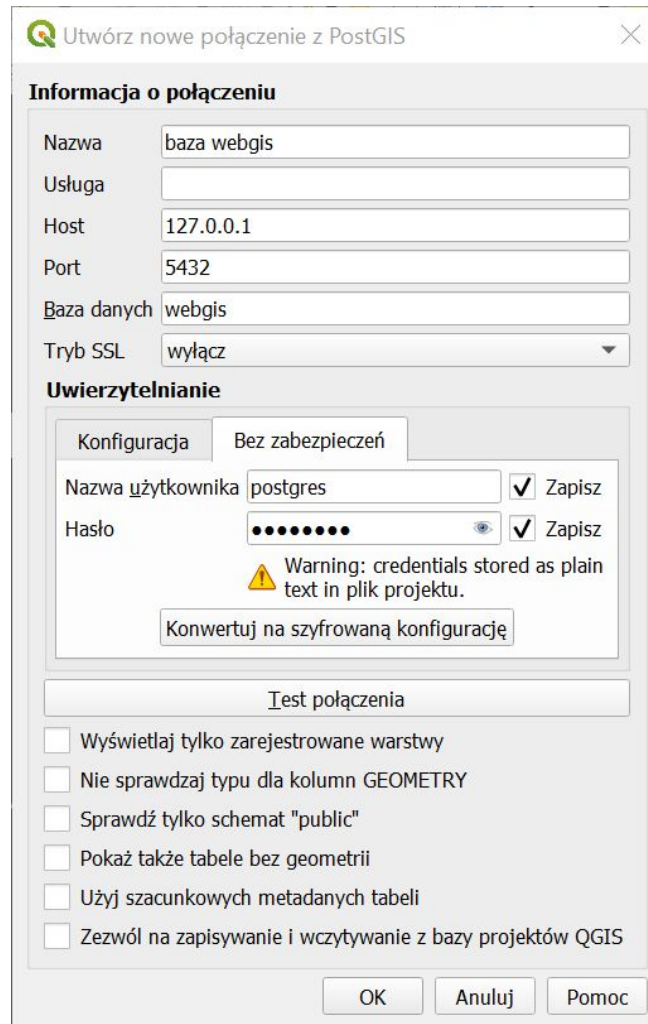
QGIS i PostGIS

QGIS wspiera bazę danych *PostgreSQL* i jej rozszerzenie przestrzenne *PostGIS* od najwcześniejszych wersji. Obecnie jest to jedno z głównych narzędzi wykorzystywanych do zarządzania i wizualizacji danych przestrzennych z tej bazy.

Łączenie z bazą danych

Aby utworzyć nowe połączenie z bazą danych należy otworzyć okno *Zarządzanie źródłami danych* wybierając menu *Warstwa -> Zarządzanie źródłami danych* lub przez odpowiednią ikonę na pasku narzędzi . W oknie dialogowym należy przejść na zakładkę *PostgreSQL*, i kliknąć *Nowe*. Następnie należy wypełnić formularz podając dane połączenia do bazy (host 127.0.0.1, port 5432, baza webgis, użytkownik postgres, hasło webgis). Podając nazwę użytkownika i hasło warto zaznaczyć opcje *Zapisz*, aby przy każdym połączeniu nie podawać tych informacji.


Klikając przycisk *Test połączenia* można zweryfikować czy podane dane są poprawne. Jeśli wszystko jest w porządku należy wcisnąć *OK*.

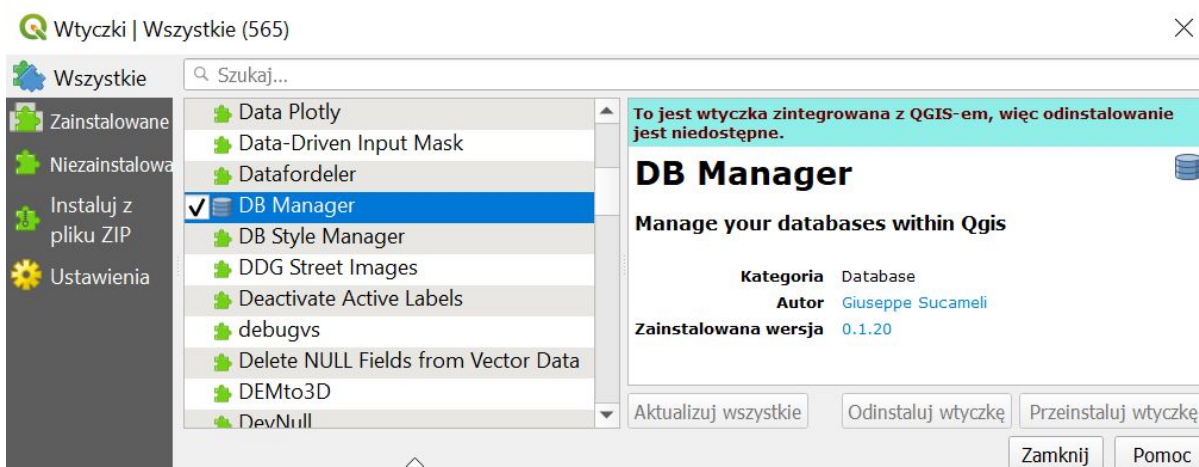


Na liście dostępnych połączeń powinna pojawić się nowa pozycja o podanej nazwie. Po kliknięciu przycisku *Połącz QGIS* pobierze dane o dostępnych schematach i tabelach w bazie. Na razie nie ma w niej żadnych danych przestrzennych więc lista będzie pusta.

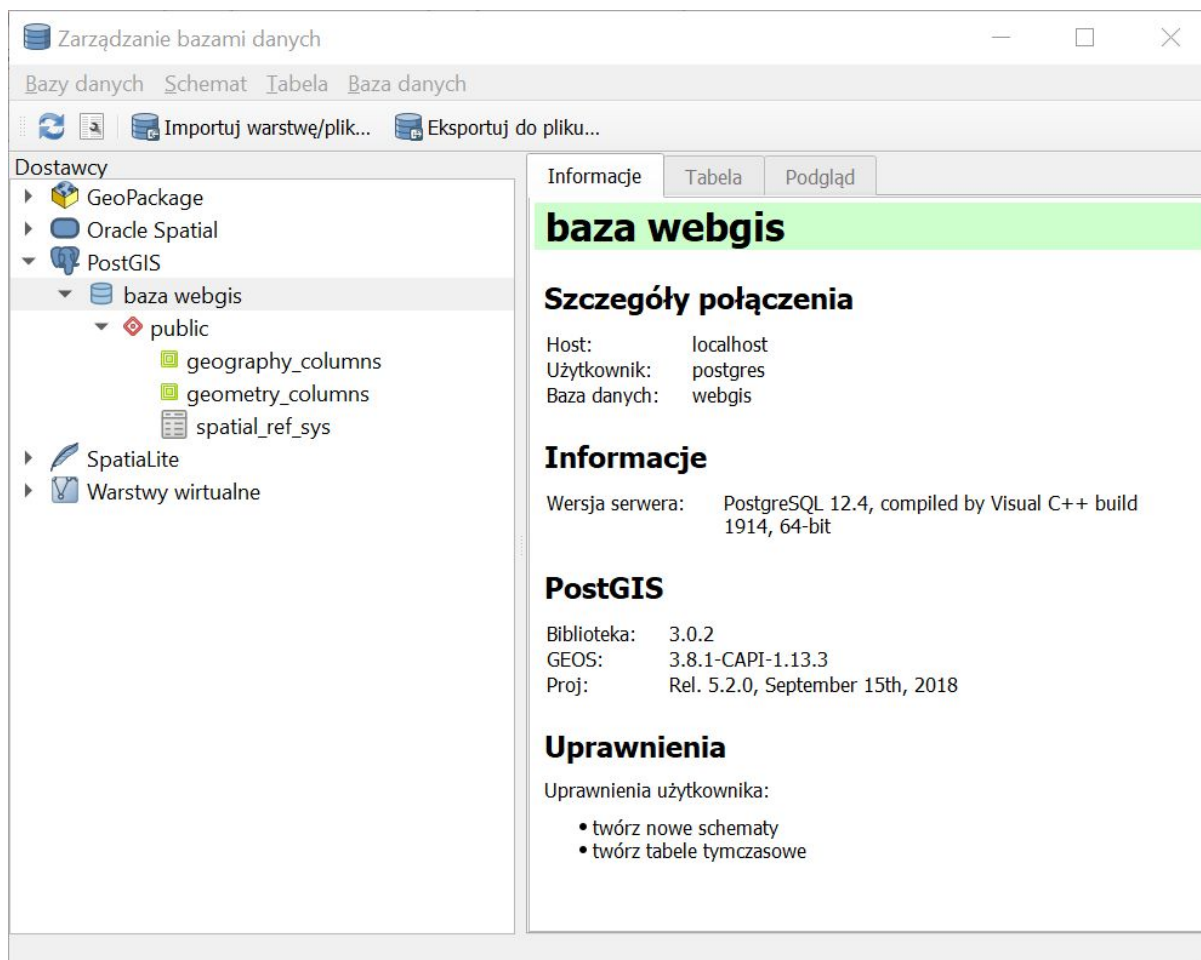
DB Manager

Wraz z QGIS instalowana jest wtyczka *DB Manager*, która pozwala na zarządzanie wspieranymi przez tą aplikację bazami danych, w tym *PostgreSQL*. Posiada ona niektóre funkcjonalności *pgAdmin* np. możliwość przeglądania schematów i tabel oraz wykonywania zapytań SQL, ale ma też dodatkowe funkcjonalności m.in. możliwość importu warstwy QGIS do bazy jako nowej tabeli, bezpośredni podgląd warstw przestrzennych czy wczytanie wyników SQL jako tabeli/warstwy do QGIS.

DB Manger jest dostępny w menu *Baza danych -> Zarządzanie bazami* danych lub poprzez przycisk  na pasku narzędzi. Jeśli wtyczka nie jest widoczna należy ją włączyć w *Menedżerze wtyczek*. W tym celu należy wejść w menu *Wtyczki -> Zarządzanie wtyczkami*, odnaleźć na liście pozycję *DB Manager* i zaznaczyć pole wyboru.



Po uruchomieniu pojawia się okno dialogowe, z prawej strony dostępne są zapisane w *QGIS* połączenia do baz danych podzielone wg ich rodzajów. Po rozwinięciu grupy *PostGIS* powinna być widoczna dodana wcześniej baza *webgis*. Rozwijając kolejne poziomy można przeglądać schematy i tabele.



Import danych wektorowych do bazy PostGIS

Korzystając z wtyczki DB Manager możliwe jest wgranie warstw QGIS do bazy *PostGIS*. W tym celu należy ją uruchomić i odnaleźć na liście pożądaną bazę i ją rozwinąć w celu połączenia. Następnie klikamy w przycisk *Importuj warstwę/plik*.

W oknie należy wskazać warstwę do wgrania, można wybrać z listy warstw wczytanych do *QGIS* lub określić ścieżkę do pliku na dysku. Następnie należy podać schemat, w którym ma być utworzona nowa tabela i jej nazwę (domyślnie jest to nazwa wybranej warstwy).

W sekcji *Opcje* można ustawić kilka elementów:

- **Klucz główny** - każdy rekord w tabeli posiada informację, która umożliwia jego jednoznaczną identyfikację, może to być pojedyncza kolumna lub ich określony zbiór. Z poziomu *DB Manger* można podać nazwę kolumny, która będzie przechowywała unikalny identyfikator, jeśli dana kolumna istnieje w warstwie to zostanie użyta, jeśli nie to powstanie nowa kolumna,
- **Kolumna geometrii** - nazwa kolumny przechowującej geometrię,
- **Źródłowy/docelowy układ współrzędnych** - określenie układu wgrywanej warstwy i docelowego układu tabeli po wgraniu danych. Jeśli wartości się różnią to dane przestrzenne zostaną transformowane do układu docelowego,
- **Kodowanie** - kodowanie znaków warstwy,
- **Nadpisz tabelę (jeśli istnieje)** - jeśli tabela o podanej nazwie istnieje w bazie to zostanie nadpisana nowymi danymi,
- **Nie zmieniaj na wieloczęściowe (multi-part)** - wymuszenie tworzenia geometrii pojedynczych (*single parts*), jeśli konwersja się nie uda zapisywanie zostanie przerwane,
- **Zamień nazwy pól na małe litery** - czy duże litery w nazwach kolumn mają być zamieniane na małe, opcja ta jest związana z faktem, że *PostgreSQL* zamienia duże litery w nazwach kolumn na ich małe odpowiedniki, jeśli kolumna ma duże litery w nazwie wymagane jest używanie w zapytaniach SQL cudzysłowów,
- **Twórz indeks przestrzenny** - automatyczne utworzenie indeksu przestrzennego dla kolumny geometrii, zazwyczaj warto zaznaczyć tą opcję,
- **Komentarz** - dodatkowy komentarz do tabeli.

Import warstwy wektorowej

Wejście: miasta

Importuj tylko zaznaczone obiekty

Tabela wyjściowa

Schemat: public

Tabela: miasta

Opcje

Klucz główny: id

Kolumna geometrii: geom

Źródłowy układ współrzędnych: EPSG:4326 - WGŚ

Docelowy układ współrzędnych: EPSG:4326 - WGŚ

Kodowanie: UTF-8

Nadpisz tabelę (jeśli istnieje)

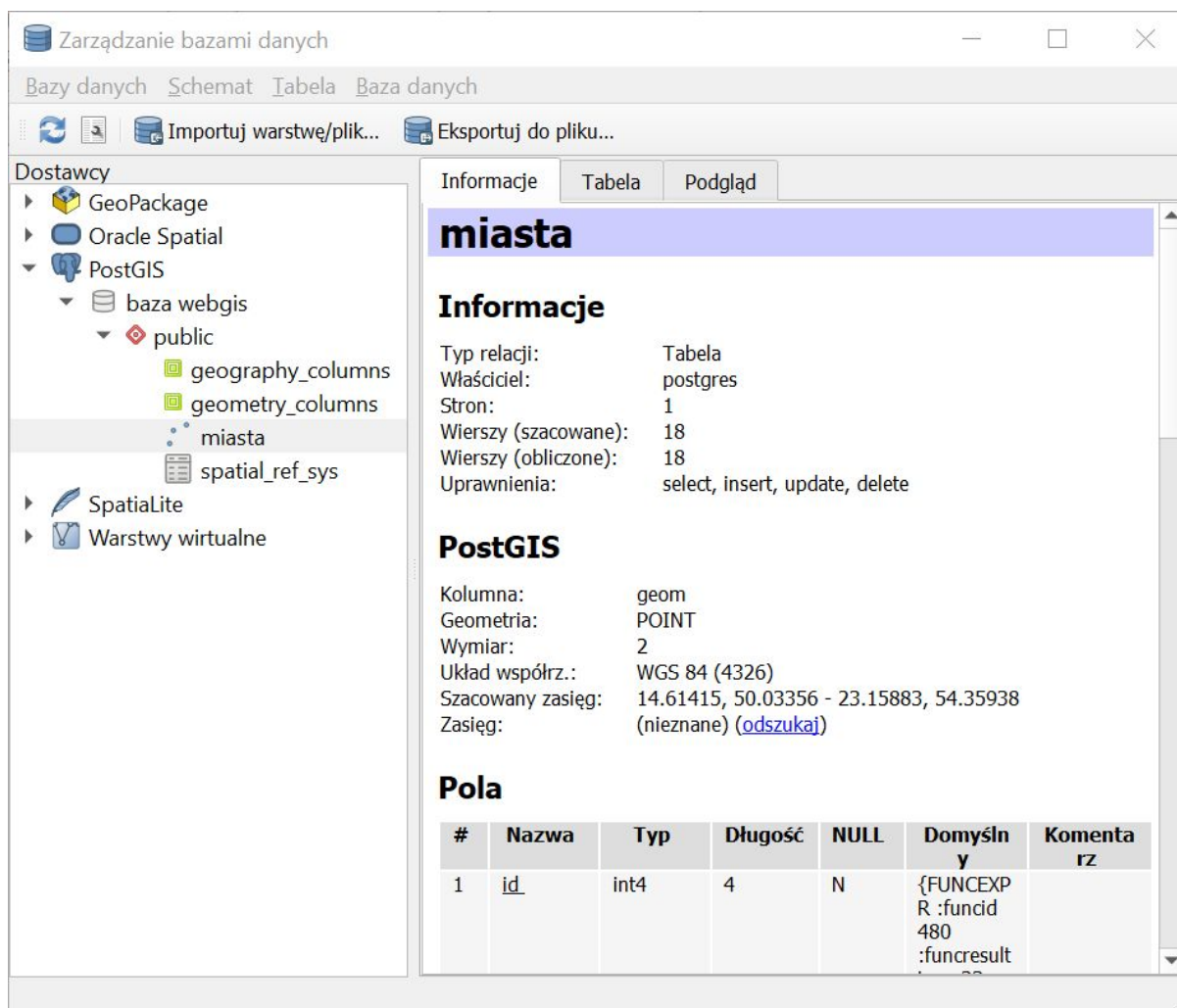
Nie zamieniaj na wieloczęściowe (multi-part)

Zamień nazwy pól na małe litery

Twórz indeks przestrzenny

Komentarz

Po ustawieniu wartości i kliknięciu *OK* tabela zostanie utworzona i wypełniona danymi. Pojawi się w odpowiednim schemacie, po jej zaznaczeniu można przejrzeć jej metadane (zakładka *Informacje*), rekordy (*Tabela*) i podejrzeć dane przestrzenne (*Podgląd*). Podwójne kliknięcie na tabeli spowoduje jej wczytanie do *QGIS*.




Ćwiczenie

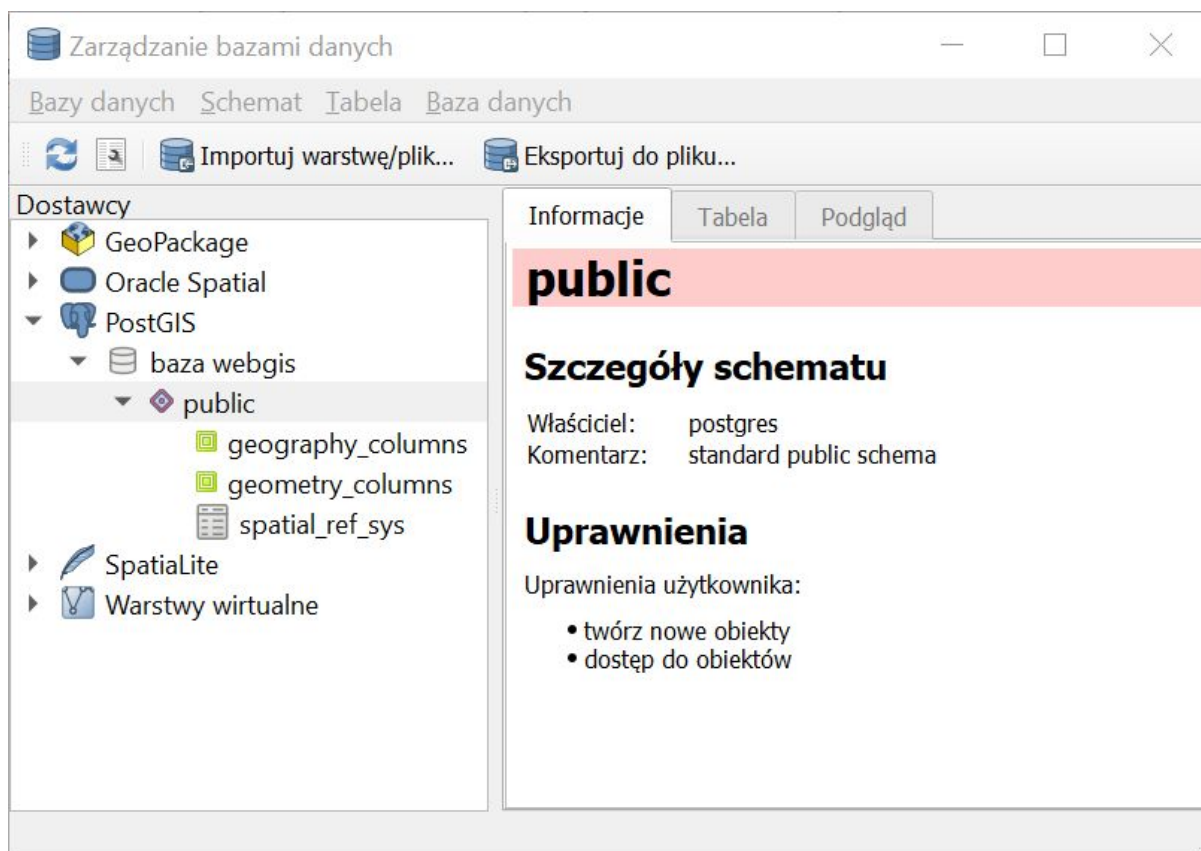
Treść zadania

Wykorzystując wtyczkę *DB Manager* dodaj do bazy danych *webgis* warstwy *miasta* i *kondracki* z testowego projektu.

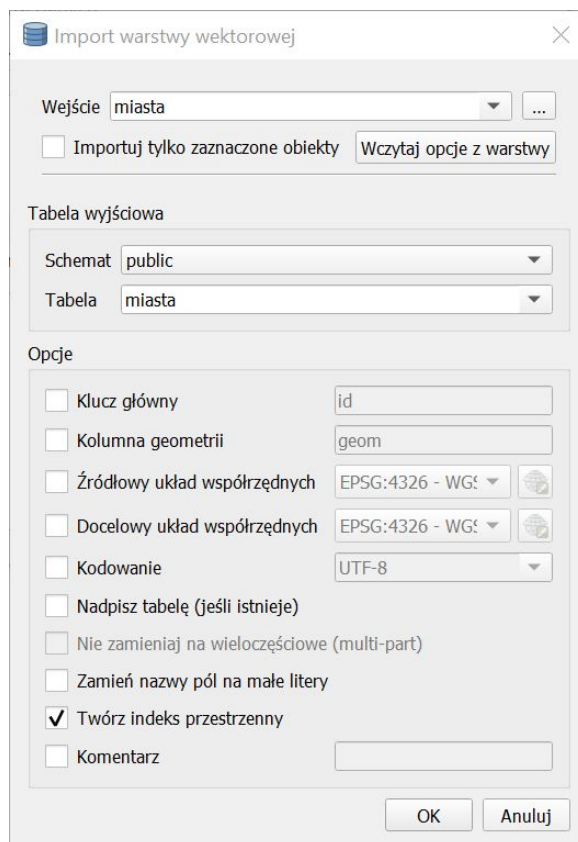
Opis

W QGIS powinny być wczytane warstwy do wgrania, więc w pierwszej kolejności wczytujemy projekt ćwiczeniowy.

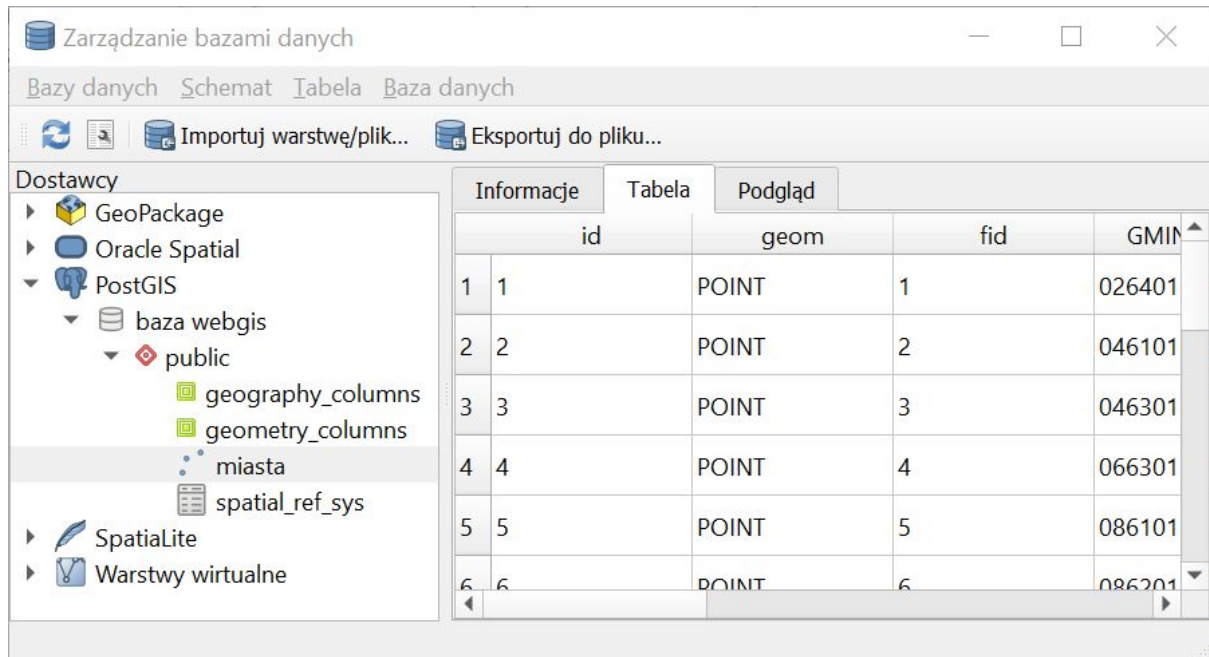
Aby uruchomić wtyczkę *DB Manger* należy w menu *Baza danych* wybrać *Zarządzanie bazami danych* lub kliknąć przycisk  na pasku narzędzi. W otworzonym oknie rozwijamy listę *PostGIS* -> *baza webgis* -> *public*.



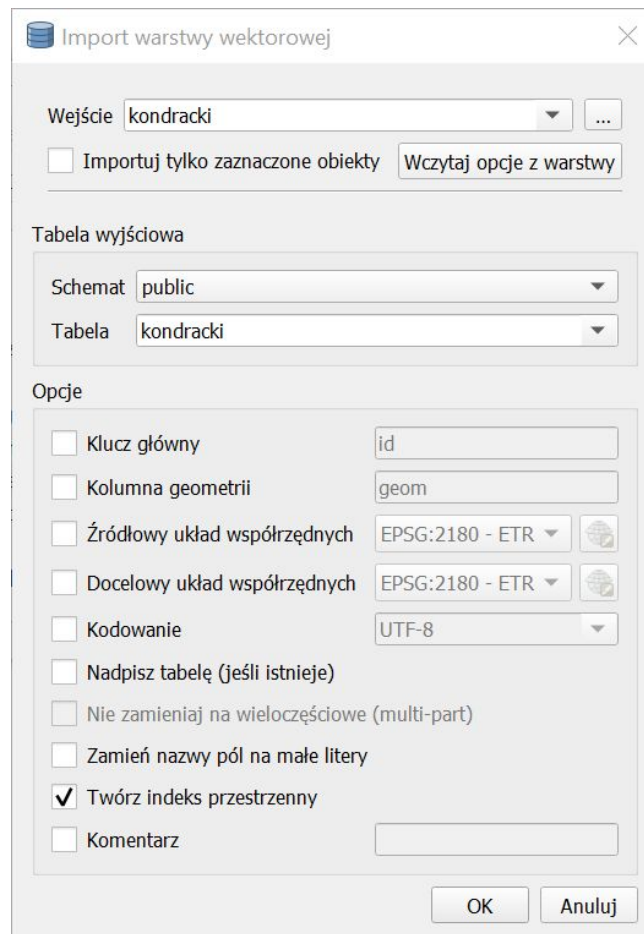
Następnie należy kliknąć przycisk *Importuj warstwę/plik*. Najpierw wgramy warstwę *miasta* pod tą samą nazwą w schemacie *public*. Z dostępnych opcji zaznaczymy *Twórz indeks przestrzenny* i klikamy OK.



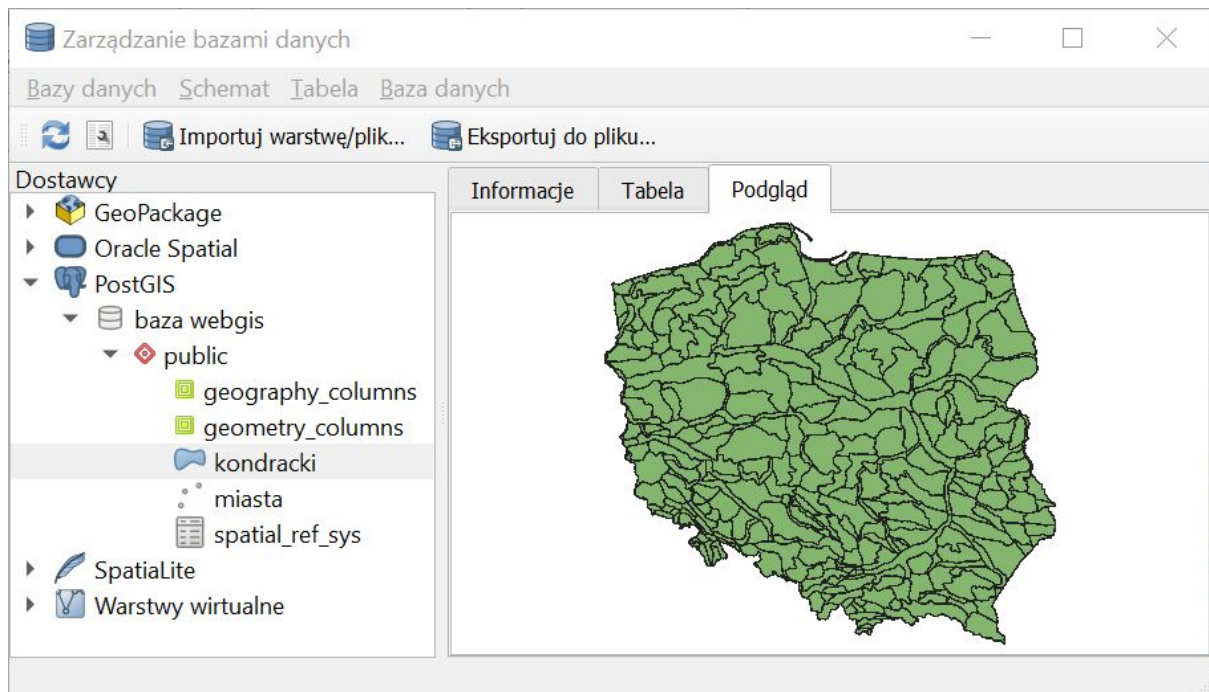
Po zakończeniu wgrwania pojawi się komunikat informujący o zakończeniu importowania danych, który zamykamy klikając na *OK*. W schemacie *public* bazy webgis powinna pojawić się nowa tabela *miasta*. Po jej zaznaczeniu można zweryfikować czy dane wgrały się poprawnie.



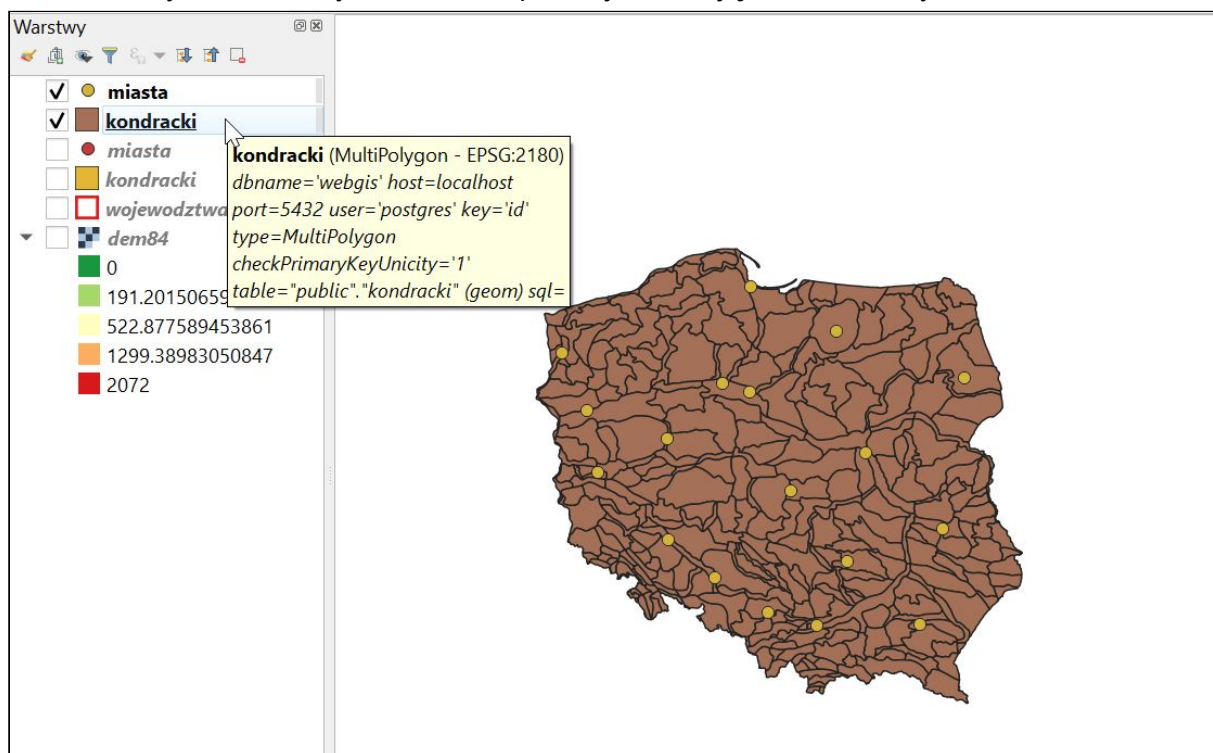
W ten sam sposób należy dodać warstwę *kondracki*.



Po zakończeniu weryfikujemy czy dane się dodały w oknie wtyczki.



Obie warstwy można wczytać do QGIS podwójnie klikając na ich nazwy.



Publikowanie danych w aplikacji Geoserver

Geoserver jest aplikacją do udostępniania danych przestrzennych przez sieć. Obsługuje ona wiele otwartych standardów OGC m.in. WMS, WFS, WMTS. Obsługuje wiele formatów GIS, zarówno wektorowych jak i rastrowych. Posiada również zestaw dodatkowych funkcjonalności m.in. grupowanie warstw, transformację danych między różnymi układami

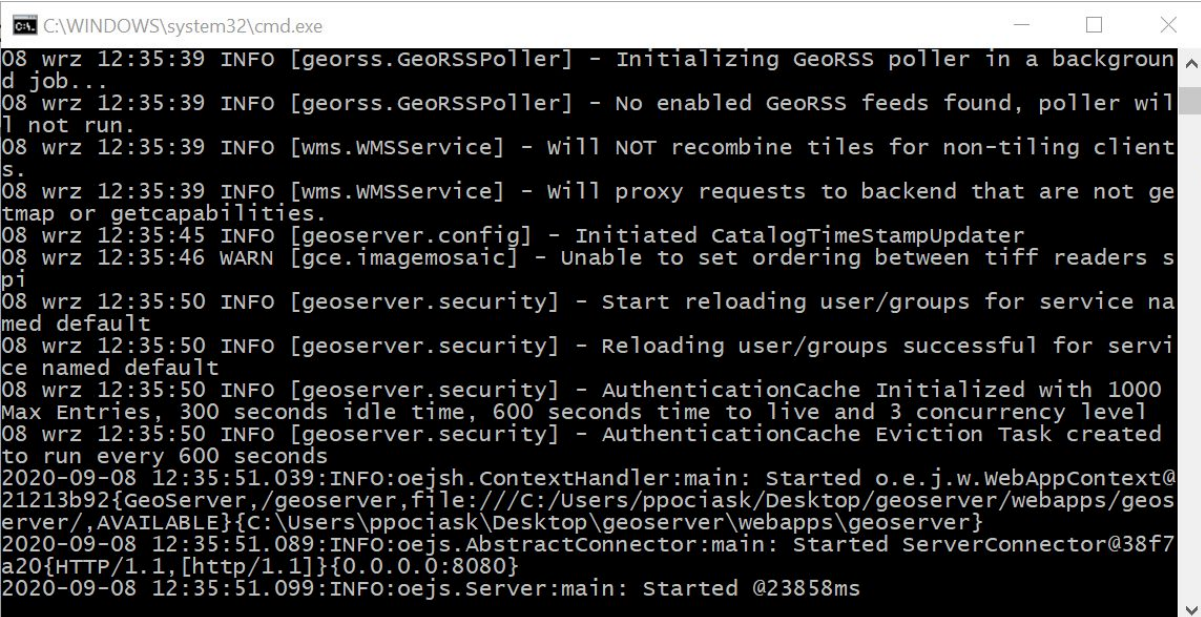
współrzędnych, symbolizację, buforowanie oraz kontrolę dostępu i uprawnień użytkowników. Dzięki systemowi rozszerzeń możliwe jest zwiększenie standardowych możliwości m.in. o obsługę nowych formatów i usług czy wsparcie modelu danych INSPIRE.

Instalacja w systemie Windows

Geoserver jest napisany w języku Java i do działania wymaga środowiska uruchomieniowego *Java Runtime Environment (JRE)*. Rekomendowana wersja to Java 8. Instalator aplikacji można pobrać ze strony głównej: <http://geoserver.org/download/>. Należy wskazać najnowszą wersję stabilną (*Stable*). Następnie wybieramy paczkę *Platform Independent Binary*. Zostanie pobrane archiwum ZIP, które należy rozpakować w dowolnym katalogu.

Pobrana wersja aplikacji dostarczana jest razem z aplikacją serwerową *Jetty* do uruchamiania sieciowych programów napisanych w języku Java. Dzięki temu nie ma potrzeby instalowania dodatkowego oprogramowania i jego konfiguracji. Istnieje możliwość użycia innych kontenerów aplikacji webowych dla języka Java np. *Apache Tomcat*, jednak do celów szkoleniowych będziemy korzystać z domyślnego rozwiązania.

Aby uruchomić aplikację za pomocą serwera *Jetty* należy wejść do stworzonego katalogu, następnie utworzyć folder *bin* i uruchomić plik *start.bat*. Aplikacja domyślnie uruchamia się na porcie 8080.



```
C:\WINDOWS\system32\cmd.exe
08 wrz 12:35:39 INFO [georss.GeorSSPoller] - Initializing GeorSS poller in a background job...
08 wrz 12:35:39 INFO [georss.GeorSSPoller] - No enabled GeorSS feeds found, poller will not run.
08 wrz 12:35:39 INFO [wms.WMSService] - will NOT recombine tiles for non-tiling clients.
08 wrz 12:35:39 INFO [wms.WMSService] - will proxy requests to backend that are not getmap or getcapabilities.
08 wrz 12:35:45 INFO [geoserver.config] - Initiated CatalogTimeStampUpdater
08 wrz 12:35:46 WARN [gce.imagemosaic] - Unable to set ordering between tiff readers spi
08 wrz 12:35:50 INFO [geoserver.security] - Start reloading user/groups for service named default
08 wrz 12:35:50 INFO [geoserver.security] - Reloading user/groups successful for service named default
08 wrz 12:35:50 INFO [geoserver.security] - AuthenticationCache Initialized with 1000 Max Entries, 300 seconds idle time, 600 seconds time to live and 3 concurrency level
08 wrz 12:35:50 INFO [geoserver.security] - AuthenticationCache Eviction Task created to run every 600 seconds
2020-09-08 12:35:51.039:INFO:oejsh.ContextHandler:main: Started o.e.j.w.WebAppContext@21213b92{GeoServer,/geoserver,file:///C:/Users/ppociask/Desktop/geoserver/webapps/geoserver/,AVAILABLE}{C:/Users/ppociask/Desktop/geoserver/webapps/geoserver}
2020-09-08 12:35:51.089:INFO:oejs.AbstractConnector:main: Started ServerConnector@38f7a20{HTTP/1.1,[http/1.1]}{0.0.0.0:8080}
2020-09-08 12:35:51.099:INFO:oejs.Server:main: Started @23858ms
```

Panel administracyjny

Panel administracyjny pozwala zarządzać wszystkimi elementami aplikacji Geoserver, w szczególności danymi przestrzennymi oraz dostępem do nich. Po uruchomieniu aplikacji serwerowej dostęp do panelu można uzyskać z poziomu przeglądarki internetowej pod adresem <http://localhost:8080/geoserver/web>. Przy pierwszym logowaniu należy podać nazwę użytkownika *admin* i hasło *geoserver*.

Panel jest częściowo spolszczony, nazwy konkretnych elementów podawane są dla wersji *Geoserver 2.17*. W innych wersjach nazwy mogą się różnić w zależności od stopnia tłumaczenia.

Z prawej strony panelu jest menu pozwalające przechodzić do ustawień aplikacji dotyczących konkretnych zagadnień.

- **About & Status** - w tej części możliwe jest sprawdzenie stanu działania *Geoserver*, przejrzanie logów oraz podanie publicznych danych kontaktowych, które są dodawane do metadanych usług.
- **Dane** - możliwe jest zarządzanie danymi przestrzennymi i ich symbolizacją.
- **Usługi** - opcje dotyczące poszczególnych usług sieciowych.
- **Ustawienia** - ustawienia aplikacji *Geoserver* m.in. logowanie i sposób renderowania obrazów.
- **Buforowanie kafelków** - opcje związane z buforowaniem (*cachowanie*) danych.
- **Bezpieczeństwo** - umożliwi zabezpieczanie dostępu do danych, tworzenie nowych użytkowników, zmianę haseł itp.



Z prawej strony widoczne są dostępne usługi. Po kliknięciu w wersję zostanie pobrany plik XML zawierający konfigurację danej usługi tzw. *Capabilities*. Służy on aplikacjom klienckim (jak *QGIS* czy *OpenLayers*) do pobrania metadanych usługi, które zawierają informacje o danym zasobie m.in. opis danych, informacje kontaktowe, listę dostępnych zasobów, formaty danych i układy współrzędnych, w jakich są one dostępne, dostępne style.

The screenshot shows the GeoServer web interface. At the top, it says 'Logged in as admin.' and has a 'Logout' button. The main heading is 'Serdecznie Witamy' (Welcome). Below this, it says 'Ten GeoServer należy do The Ancient Geographers.' There are three rows of server statistics: '23 Warstwy' (Layers) with an 'Add layers' button, '12 Stores' with an 'Add stores' button, and '8 Workspaces' with a 'Create workspaces' button. There are two warning messages: 'Domyślne hasło głównego serwera nie zostało zmienione. Jest **wysoce** rekomendowane, aby zmienić je teraz. Zmień je' and 'Domyślne hasło administratora serwera nie zostało zmienione. Jest **wysoce** rekomendowane, aby zmienić je teraz. Zmień je'. There is also an information icon and the text 'Silna kryptografia dostępna'. At the bottom, it says 'Ten GeoServer jest uruchomiony w wersji 2.17.2. Aby uzyskać więcej informacji skontaktuj się z administratorem.' On the right side, there is a section 'Możliwości usługi' (Service Capabilities) listing various services and their versions: WCS (1.0.0, 1.1.0, 1.1.1, 2.0.1), WFS (1.0.0, 1.1.0, 2.0.0), WMS (1.1.1, 1.3.0), TMS (1.0.0), WMS-C (1.1.1), and WMTS (1.0.0). On the left side, there is a navigation menu with sections: 'About & Status' (Server Status, Dzienniki GeoServer, Informacje o kontakcie, O GeoServer), 'Dane' (Podgląd warstw, Workspaces, Stores, Warstwy, Grupa warstw, Style), 'Usługi' (WMTS, WCS, WFS, WMS), and 'Ustawienia' (Global, JAI, Coverage Access).













W niektórych widokach dostępna jest główna tabela, z poziomu której możliwe jest zarządzanie danym elementem systemu. Nad tabelą dostępne opcje pozwalające dodać nowy element lub usunąć wybrane rekordy (wskazuje się je zaznaczając pole wyboru z lewej strony rekordu tabeli). Dostępne jest też pole wyszukiwania oraz przyciski pozwalające wyświetlać kolejne elementy, jeżeli wszystkie nie mieszczą się na stronie. Aby edytować dany element należy kliknąć na jego nazwę w tabeli. Rekordy można sortować klikając na nagłówkach tabel.

Warstwy

Zarządzanie warstwami opublikowanymi przez GeoServer

 Dodaj nowy zasób
 Usuń wybrane zasoby

<< < 1 > >> Results 1 to 19 (out of 19 items)

<input type="checkbox"/>	Typ	Title	Nazwa warstwy	Magazyn	Wiączony?	Natywna SRS
<input type="checkbox"/>		World rectangle	tiger:giant_polygon	nyc	<input checked="" type="checkbox"/>	EPSG:4326
<input type="checkbox"/>		Manhattan (NY) points of interest	tiger:poi	nyc	<input checked="" type="checkbox"/>	EPSG:4326
<input type="checkbox"/>		Manhattan (NY) landmarks	tiger:poly_landmarks	nyc	<input checked="" type="checkbox"/>	EPSG:4326
<input type="checkbox"/>		Manhattan (NY) roads	tiger:tiger_roads	nyc	<input checked="" type="checkbox"/>	EPSG:4326
<input type="checkbox"/>		A sample ArcGrid file	nurc:Arc_Sample	arcGridSample	<input checked="" type="checkbox"/>	EPSG:4326
<input type="checkbox"/>		North America sample imagery	nurc:Img_Sample	worldImageSample	<input checked="" type="checkbox"/>	EPSG:4326
<input type="checkbox"/>		Pk50095	nurc:Pk50095	img_sample2	<input checked="" type="checkbox"/>	EPSG:32633
<input type="checkbox"/>		mosaic	nurc:mosaic	mosaic	<input checked="" type="checkbox"/>	EPSG:4326
<input type="checkbox"/>		USA Population	topp:states	states_shapefile	<input checked="" type="checkbox"/>	EPSG:4326
<input type="checkbox"/>		Tasmania cities	topp:tasmania_cities	taz_shapes	<input checked="" type="checkbox"/>	EPSG:4326
<input type="checkbox"/>		Tasmania roads	topp:tasmania_roads	taz_shapes	<input checked="" type="checkbox"/>	EPSG:4326
<input type="checkbox"/>		Tasmania state boundaries	topp:tasmania_state_boundaries	taz_shapes	<input checked="" type="checkbox"/>	EPSG:4326

Zarządzanie danymi przestrzennymi

Zarządzanie danymi odbywa się w sekcji *Dane* panelu administracyjnego. Dostępnych jest kilka podstron, w których definiuje się konkretne elementy związane z danymi przestrzennymi w aplikacji *Geoserver*:

- **Podgląd warstw** - lista wszystkich warstw z możliwością ich podglądu w różnych formatach,
- **Workspaces** - zarządzanie obszarami roboczymi,
- **Stores** - zarządzanie źródłami danych,
- **Warstwy** - zarządzanie warstwami,
- **Grupa warstw** - umożliwia grupowanie warstw i traktowanie ich jak pojedyncza warstwa,
- **Style** - tworzenie stylów, które mogą być przypisane warstwom.

Obszary robocze (workspaces)

Workspaces

Manage GeoServer workspaces

Add new workspace

Remove selected workspace(s)

<< < 1 > >> Results 1 to 7 (out of 7 items) Search

<input type="checkbox"/>	Workspace Name	Domyślne	Isolated
<input type="checkbox"/>	cite		
<input type="checkbox"/>	it.geosolutions		
<input type="checkbox"/>	nurc		
<input type="checkbox"/>	sde		
<input type="checkbox"/>	sf		
<input type="checkbox"/>	tiger		
<input type="checkbox"/>	topp		

<< < 1 > >> Results 1 to 7 (out of 7 items)

Podstawowym elementem grupującym warstwy jest obszar roboczy. Jest to kontener pozwalający organizować inne elementy w systemie. Dostęp do warstw może odbywać się poprzez podanie nazwy obszaru roboczego i warstwy rozdzielonych dwukropkiem np. `obszar_robotczy:warstwa`. Dwie warstwy w *Geoserver* mogą mieć taką samą nazwę tylko jeśli znajdują się w innych obszarach roboczych.

Aby dodać nowy obszar roboczy należy kliknąć *Add new workspace*. W nowym oknie należy podać dane dla nowego obszaru:

- **Name** - nazwa obszaru, może zawierać litery, liczby i znaki `.-_`. Pozostałe znaki (m.in. spacje) są niedozwolone,
- **Namespace URI** - przestrzeń nazw, określa unikalny identyfikator obszaru roboczego, zazwyczaj jest tworzony poprzez podanie domeny głównej aplikacji wraz z nazwą obszaru roboczego np. <https://strona.pl/workspace>,
- **Default workspace** - czy obszar roboczy jest domyślny, tylko jeden workspace może być domyślny, są do niego przypisane wszystkie elementy systemu, które nie mają określonego innego obszaru roboczego,
- **Isolated workspace** - dane z izolowanych obszarów roboczych nie są widoczne przy globalnym pobieraniu informacji o zasobach *Geoservera*, aby mieć do nich dostęp należy w adresie określić nazwę obszaru roboczego. Dzięki izolowaniu możliwe jest również tworzenie więcej niż jednego obszaru roboczego w tej samej przestrzeni nazw (ta sama wartość dla *Namespace URI*). W takim wypadku jednocześnie tylko jeden obszar może być nieizolowany, pozostałe muszą być oznaczone jako izolowane.

New Workspace

Configure a new workspace

Basic Info **Security**

Name

Namespace URI

The namespace uri associated with this workspace

Default Workspace

Isolated Workspace

Wyslij **Cancel**

W zakładce *Security* możliwe jest określenie uprawnień do obszaru roboczego poszczególnych użytkowników.

Ćwiczenie

Treść zadania

Dodaj nowy obszar roboczy o nazwie *webgis* i przestrzeni nazw <http://localhost:8080/webgis>.

Opis

W panelu administracyjnym *Geoserver* należy w sekcji *Dane* wybrać *Workspace*. Będąc na stronie do zarządzania obszarami roboczymi klikamy przycisk *Add new workspace*.

The screenshot shows the GeoServer administration interface. On the left, the 'Workspaces' menu item is highlighted with a red box and the number '1'. In the main content area, the 'Add new workspace' button is highlighted with a red box and the number '2'. Below the button, there is a list of existing workspaces: 'cite', 'it.geosolutions', 'nurc', and 'sde'. The 'Add new workspace' button is a green circle with a plus sign, and the 'Remove selected workspace(s)' button is a red circle with a minus sign. The main content area also shows navigation buttons (left and right arrows) and a pagination indicator 'Results 1 to 7 (out of 7 items)'.

W polu tekstowym *Name* należy wpisać nazwę `webgis`, a w pole *Namespace URI* `http://localhost:8080/webgis`. Pozostałe opcje pozostawiamy odznaczone.

The 'New Workspace' form is shown with the 'Basic Info' tab selected. The 'Name' field contains the text 'webgis'. The 'Namespace URI' field contains the text 'http://localhost:8080/webgis'. Below the 'Namespace URI' field, there is a note: 'The namespace uri associated with this workspace'. There are two checkboxes: 'Default Workspace' and 'Isolated Workspace', both of which are unchecked. At the bottom of the form, there are two buttons: 'Wyślij' (Send) and 'Cancel'.

Po kliknięciu *Wyślij* pojawi się tabela, w której będzie widoczny utworzony obszar roboczy.

Workspaces

Manage GeoServer workspaces

Results 1 to 8 (out of 8 items)

<input type="checkbox"/>	Workspace Name
<input type="checkbox"/>	cite
<input type="checkbox"/>	it.geosolutions
<input type="checkbox"/>	nurc
<input type="checkbox"/>	sde
<input type="checkbox"/>	sf
<input type="checkbox"/>	tiger
<input type="checkbox"/>	topp
<input type="checkbox"/>	webgis

Results 1 to 8 (out of 8 items)

Magazyny danych (stores)

Stores

Manage the stores providing data to GeoServer

Results 1 to 9 (out of 9 items)

<input type="checkbox"/>	Typ	Obszar roboczy	Store Name	Typ	Włączony?
<input type="checkbox"/>		nurc	arcGridSample	ArcGrid	✓
<input type="checkbox"/>		nurc	img_sample2	WorldImage	✓
<input type="checkbox"/>		nurc	mosaic	ImageMosaic	✓
<input type="checkbox"/>		tiger	nyc	Shapefile	✓
<input type="checkbox"/>		sf	sf	Shapefile	✓
<input type="checkbox"/>		sf	sfdem	GeoTIFF	✓
<input type="checkbox"/>		topp	states_shapefile	Shapefile	✓
<input type="checkbox"/>		topp	taz_shapes	Shapefile	✓
<input type="checkbox"/>		nurc	worldImageSample	WorldImage	✓

Results 1 to 9 (out of 9 items)

Magazyny danych pozwalają na bezpośrednią komunikację ze źródłem danych np. z bazą danych lub plikiem na dysku i pobieranie z nich informacji przestrzennych. Dzięki magazynom możliwe jest określenie połączenia z konkretnym źródłem danych raz (np. z bazą danych) i późniejsze dodanie wielu warstw z tego źródła (np. różne tabele w bazie). Możliwe jest dodanie danych wektorowych, rastrowych oraz innych źródeł usług OGC WMS/WMFS. W tym ostatnim przypadku Geoserver działa jako *proxy*, czyli pośredniczy w

przekazywaniu danych z innych serwisów. W nomenklaturze Geoserver określa się takie usługi jako kaskadowe.

Aby dodać nowy magazyn należy kliknąć *Add new Store*.

Nowe źródło danych

Wybierz typ źródła danych, który chcesz skonfigurować

Wektorowe Źródła Danych

- Directory of spatial files (shapefiles) - Takes a directory of shapefiles and exposes it as a data store
- GeoPackage - GeoPackage
- PostGIS - PostGIS Database
- PostGIS (JNDI) - PostGIS Database (JNDI)
- Properties - Allows access to Java Property files containing Feature information
- Shapefile - ESRI(tm) Shapefiles (*.shp)
- Web Feature Server (NG) - Provides access to the Features published a Web Feature Service, and the ability to perform transactions on the server (when supported / allowed).

Rastrowe Źródła Danych

- ArcGrid - ARC/INFO ASCII GRID Coverage Format
- GeoPackage (mosaic) - GeoPackage mosaic plugin
- GeoTIFF - Tagged Image File Format with Geographic information
- ImageMosaic - Image mosaicking plugin
- WorldImage - A raster file accompanied by a spatial data file

Inne Źródła Danych

- WMS - Kaskaduje zdalny Web Map Service
- WMTS - Cascades a remote Web Map Tile Service

Następnie należy wskazać jaki typ danych będzie dodawany.

Nowe źródło danych wektorowych

Dodaj nowe źródło danych wektorowych

PostGIS
PostGIS Database

Basic Store Info

Obszar roboczy *

cite

Nazwa źródła danych *

Opis

Włączone

Connection Parameters

host *

localhost

port *

5432

database

schema

public

user *

passwd

.....

Dane w części *Basic Store Info* są wspólne dla wszystkich formatów:

- **Obszar roboczy** - określa obszar roboczy, do którego zostanie dodany magazyn danych,
- **Nazwa źródła danych** - nazwa widoczna m.in. na liście,
- **Opis** - dłuższy tekst opisujący źródło danych (opcjonalne),
- **Włączone** - jeśli ta opcja jest wyłączona to wszystkie warstwy zdefiniowane z tego źródła danych nie są widoczne w usługach.

W części *Connection parameters* określone są dane potrzebne do pobrania informacji z danego typu źródła. Każdy rodzaj ma swoje parametry np. dla bazy *PostgreSQL* należy podać parametry połączenia (host, port, nazwa bazy i dane użytkownika), a dla danych plikowych (*ESRI Shapefile*, *GeoTIFF*) ich lokalizację na dysku.

Po zapisaniu zmian istnieje możliwość publikacji danych z nowego źródła w formie warstwy.

Ćwiczenie

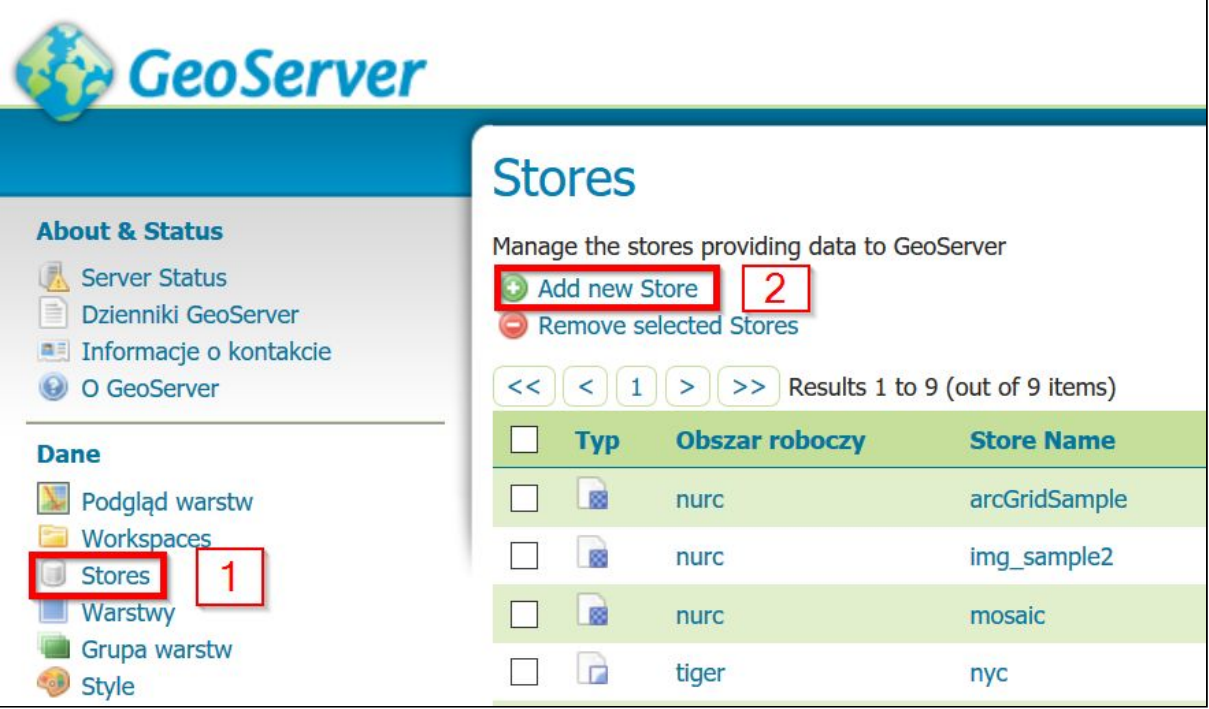
Treść zadania

Dodaj trzy nowe magazyny danych do obszaru roboczego *webgis* dla:

1. rastra *dem84.tif*,
2. pliku ESRI Shapefile *województwa.shp*,
3. bazy danych *webgis*.

Opis

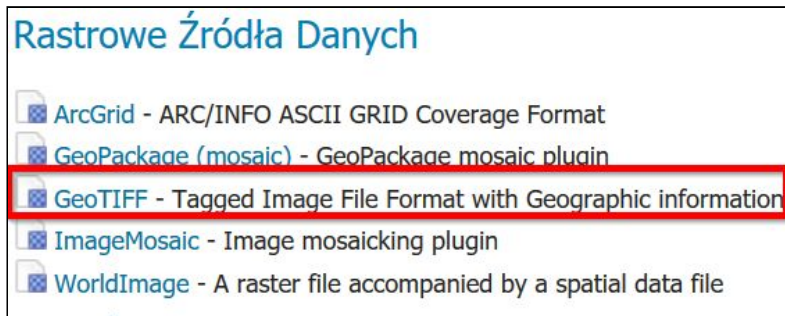
W panelu administracyjnym Geoserver należy w sekcji *Dane* wybrać *Stores*. Będąc na stronie do zarządzania magazynami danych klikamy przycisk *Add new store*.



The screenshot shows the GeoServer administration interface. On the left sidebar, under the 'Dane' section, the 'Stores' link is highlighted with a red box and the number '1'. In the main content area, the 'Stores' management page is displayed. At the top, there is a red box around the 'Add new Store' button and another red box with the number '2' next to it. Below the buttons, there is a table of existing stores:

<input type="checkbox"/>	Typ	Obszar roboczy	Store Name
<input type="checkbox"/>		nurc	arcGridSample
<input type="checkbox"/>		nurc	img_sample2
<input type="checkbox"/>		nurc	mosaic
<input type="checkbox"/>		tiger	nyc

W celu dodania rastra z pliku z części *Rastrowe Źródła Danych* należy wybrać jeden ze wspieranych formatów. Plik *dem84.tif* jest w formacie *GeoTIFF*.



Na nowej stronie w części *Basic Store Info* wybieramy obszar roboczy webgis i podajemy nazwę dla źródła danych np. *nmt_store*. W sekcji *Connection Parameters* należy wskazać lokalizację pliku. W tym celu można wpisać ścieżkę ręcznie (z przedrostkiem *file://*) lub kliknąć przycisk *Przeglądaj* aby wybrać plik z dysku.

Dodaj źródło danych rastrowych

Opis

GeoTIFF
Tagged Image File Format with Geographic information

Basic Store Info

Obszar roboczy *

webgis

Nazwa źródła danych *

nmt_store

Opis

Włączone

Connection Parameters

URL *

file://C:\Users\...\Desktop\dane_webgis\dem84.tif

Po wypełnieniu danych klikamy *Save*. Zostaniemy przeniesieni do strony, z której od razu można stworzyć warstwę klikając przycisk *Publish*, ale tym zajmiemy się w kolejnym ćwiczeniu. Teraz należy wrócić do listy magazynów i sprawdzić czy pojawiła się nowa pozycja. Powinna ona znajdować się w obszarze roboczym *webgis*.

Stores

Manage the stores providing data to GeoServer

Add new Store
 Remove selected Stores

Results 1 to 10 (out of 10 items)

<input type="checkbox"/>	Typ	Obszar roboczy	Store Name	Typ	Włączony?
<input type="checkbox"/>		nurc	arcGridSample	ArcGrid	<input checked="" type="checkbox"/>
<input type="checkbox"/>		nurc	img_sample2	WorldImage	<input checked="" type="checkbox"/>
<input type="checkbox"/>		nurc	mosaic	ImageMosaic	<input checked="" type="checkbox"/>
<input type="checkbox"/>		webgis	nmt_store	GeoTIFF	<input checked="" type="checkbox"/>
<input type="checkbox"/>		tiger	nyc	Shapefile	<input checked="" type="checkbox"/>
<input type="checkbox"/>		sf	sf	Shapefile	<input checked="" type="checkbox"/>
<input type="checkbox"/>		sf	sfdem	GeoTIFF	<input checked="" type="checkbox"/>
<input type="checkbox"/>		topp	states_shapefile	Shapefile	<input checked="" type="checkbox"/>
<input type="checkbox"/>		topp	taz_shapes	Shapefile	<input checked="" type="checkbox"/>
<input type="checkbox"/>		nurc	worldImageSample	WorldImage	<input checked="" type="checkbox"/>

Kolejnym krokiem jest dodanie pliku ESRI Shapefile, klikamy więc ponownie w *Add new Store*. W części *Wektorowe Źródła Danych* wybieramy opcję *Shapefile*.

Wektorowe Źródła Danych

- Directory of spatial files (shapefiles) - Takes a directory of shapefiles and exposes it as a data store
- GeoPackage - GeoPackage
- PostGIS - PostGIS Database
- PostGIS (JNDI) - PostGIS Database (JNDI)
- Properties - Allows access to Java Property files containing Feature information
- Shapefile - ESRI(tm) Shapefiles (*.shp)
- Web Feature Server (NG) - Provides access to the Features published a Web Feature Service, and the the server (when supported / allowed).

Sekcję *Basic Store Info* wypełniamy podobnie jak w przypadku rastra, podając inną nazwę źródła np. *województwa_store*. W części *Connection Parameters* należy wskazać ścieżkę do pliku oraz wybrać odpowiednie kodowanie danych w tabeli atrybutów. W przypadku pliku *województwa.shp* jest to UTF-8. Pozostałe opcje można pozostawić domyślne.

Nowe źródło danych wektorowych

Dodaj nowe źródło danych wektorowych

Shapefile

ESRI(tm) Shapefiles (*.shp)

Basic Store Info

Obszar roboczy *

webgis

Nazwa źródła danych *

województwa_store

Opis

Włączone

Connection Parameters

Shapefile location *

file:///C:/Users/.../Desktop/dane_webgis/województwa.shp

DBF charset

UTF-8

Create spatial index if missing/outdated

Użyj buforów mapowanych na pamięć (Wyłącz w Windows)

Buforuj i reużywaj map pamięci (Wymaga opcji 'Użyj buforów mapowanych na pamięć' aby być włączone)

Klikamy przycisk Save i wracamy do listy magazynów, gdzie pojawiła się nowa pozycja.

<input type="checkbox"/>	Typ	Obszar roboczy	Store Name	Typ	Włączony?
<input type="checkbox"/>		nurc	arcGridSample	ArcGrid	<input checked="" type="checkbox"/>
<input type="checkbox"/>		nurc	img_sample2	WorldImage	<input checked="" type="checkbox"/>
<input type="checkbox"/>		nurc	mosaic	ImageMosaic	<input checked="" type="checkbox"/>
<input type="checkbox"/>		webgis	nmt_store	GeoTIFF	<input checked="" type="checkbox"/>
<input type="checkbox"/>		tiger	nyc	Shapefile	<input checked="" type="checkbox"/>
<input type="checkbox"/>		sf	sf	Shapefile	<input checked="" type="checkbox"/>
<input type="checkbox"/>		sf	sfdem	GeoTIFF	<input checked="" type="checkbox"/>
<input type="checkbox"/>		topp	states_shapefile	Shapefile	<input checked="" type="checkbox"/>
<input type="checkbox"/>		topp	taz_shapes	Shapefile	<input checked="" type="checkbox"/>
<input type="checkbox"/>		webgis	województwa_store	Shapefile	<input checked="" type="checkbox"/>
<input type="checkbox"/>		nurc	worldImageSample	WorldImage	<input checked="" type="checkbox"/>

Ostatnim magazynem jest baza danych *PostGIS*. Klikamy na *Add new Store* i w części *Wektorowe Źródła Danych* wybieramy pozycję *PostGIS*.

Wektorowe Źródła Danych

- Directory of spatial files (shapefiles) - Takes a directory of shapefiles and exposes it as a data store
- GeoPackage - GeoPackage
- PostGIS - PostGIS Database**
- PostGIS (JNDI) - PostGIS Database (JNDI)
- Properties - Allows access to Java Property files containing Feature information
- Shapefile - ESRI(tm) Shapefiles (*.shp)
- Web Feature Server (NG) - Provides access to the Features published a Web Feature Service, and the the server (when supported / allowed).

Część *Basic Store Info* wypełniamy analogicznie jak przy poprzednich magazynach, nazwę źródła wpisujemy jako `postgis_store`. W części *Connection Parameters* należy podać dane wymagane do połączenia z bazą, zgodnie z poniższym przykładem. Pozostałe opcje pozostawiamy niezmienione.

Nowe źródło danych wektorowych

Dodaj nowe źródło danych wektorowych

PostGIS
PostGIS Database

Basic Store Info

Obszar roboczy *

webgis

Nazwa źródła danych *

postgis_store

Opis

Włączone

Connection Parameters

host *

localhost

port *

5432

database

webgis

schema

public

user *

postgres

passwd

••••••••

Klikamy Save. Teraz na liście widać dwie pozycje - przestrzenne tabele w bazie *webgis*.

Published	Layer name	Action
	kondracki	Publish
	miasta	Publish

Po powrocie do listy magazynów mamy trzecią nową pozycję.

<input type="checkbox"/>	Typ	Obszar roboczy	Store Name	Typ	Włączony?
<input type="checkbox"/>		nurc	arcGridSample	ArcGrid	✓
<input type="checkbox"/>		nurc	img_sample2	WorldImage	✓
<input type="checkbox"/>		nurc	mosaic	ImageMosaic	✓
<input type="checkbox"/>		webgis	nmt_store	GeoTIFF	✓
<input type="checkbox"/>		tiger	nyc	Shapefile	✓
<input type="checkbox"/>		webgis	postgis_store	PostGIS	✓
<input type="checkbox"/>		sf	sf	Shapefile	✓
<input type="checkbox"/>		sf	sfdem	GeoTIFF	✓
<input type="checkbox"/>		topp	states_shapefile	Shapefile	✓
<input type="checkbox"/>		topp	taz_shapes	Shapefile	✓
<input type="checkbox"/>		webgis	województwa_store	Shapefile	✓
<input type="checkbox"/>		nurc	worldImageSample	WorldImage	✓

Warstwy (layers)

Warstwy

Zarządzanie warstwami opublikowanymi przez GeoServer

+ Dodaj nowy zasób
- Usuń wybrane zasoby

<< < 1 > >> Results 1 to 19 (out of 19 items)

 Search

<input type="checkbox"/>	Typ	Title	Nazwa warstwy	Magazyn	Włączony?	Natywna SRS
<input type="checkbox"/>		World rectangle	tiger:giant_polygon	nyc	✓	EPSG:4326
<input type="checkbox"/>		Manhattan (NY) points of interest	tiger:poi	nyc	✓	EPSG:4326
<input type="checkbox"/>		Manhattan (NY) landmarks	tiger:poly_landmarks	nyc	✓	EPSG:4326
<input type="checkbox"/>		Manhattan (NY) roads	tiger:tiger_roads	nyc	✓	EPSG:4326
<input type="checkbox"/>		A sample ArcGrid file	nurc:Arc_Sample	arcGridSample	✓	EPSG:4326
<input type="checkbox"/>		North America sample imagery	nurc:Img_Sample	worldImageSample	✓	EPSG:4326
<input type="checkbox"/>		Pk50095	nurc:Pk50095	img_sample2	✓	EPSG:32633
<input type="checkbox"/>		mosaic	nurc:mosaic	mosaic	✓	EPSG:4326
<input type="checkbox"/>		USA Population	topp:states	states_shapefile	✓	EPSG:4326
<input type="checkbox"/>		Tasmania cities	topp:tasmania_cities	taz_shapes	✓	EPSG:4326
<input type="checkbox"/>		Tasmania roads	topp:tasmania_roads	taz_shapes	✓	EPSG:4326
<input type="checkbox"/>		Tasmania state boundaries	topp:tasmania_state_boundaries	taz_shapes	✓	EPSG:4326

Warstwa reprezentuje zbiór danych przestrzennych i pobiera informacje z połączonego z nią źródła danych. Są to podstawowe elementy w *Geoserver*, które użytkownik może

wyświetlić lub pobrać. Na głównej liście widoczne są warstwy wraz z informacjami o obszarze roboczym, magazynie i natywnym układzie współrzędnych.

Aby dodać nową warstwę należy kliknąć *Dodaj nowy zasób*. W kolejnym kroku na liście wskazujemy magazyn definiujący źródło danych. Po wyborze pojawi się lista dostępnych warstw np. tabel w bazie danych. Aby opublikować wybraną warstwę należy kliknąć przycisk *Publish*. Jeżeli warstwa jest już opublikowana widoczny jest przycisk *Publish again*. Jedną warstwę można opublikować wielokrotnie np. z innym stylem lub uprawnieniami.

Przy dodawaniu i edycji warstwy opcje podzielone są na zakładki:

- **Dane** - określenie metadanych warstwy m.in. jej nazwy i opisu, układy współrzędnych (źródła oraz do publikacji), granice danych oraz atrybutów i filtra dla obiektów wektorowych,
- **Publishing** - ustawienia dla usług *WMS/WFS/WCS*,
- **Dimensions** - opcje dla dodatkowych wymiarów w usłudze *WMS*: czas i wysokość.
- **Buforowanie kafelków** - ustawienia kafelkowania danych m.in. ich formatów i układów,
- **Security** - określenie uprawnień do warstwy.

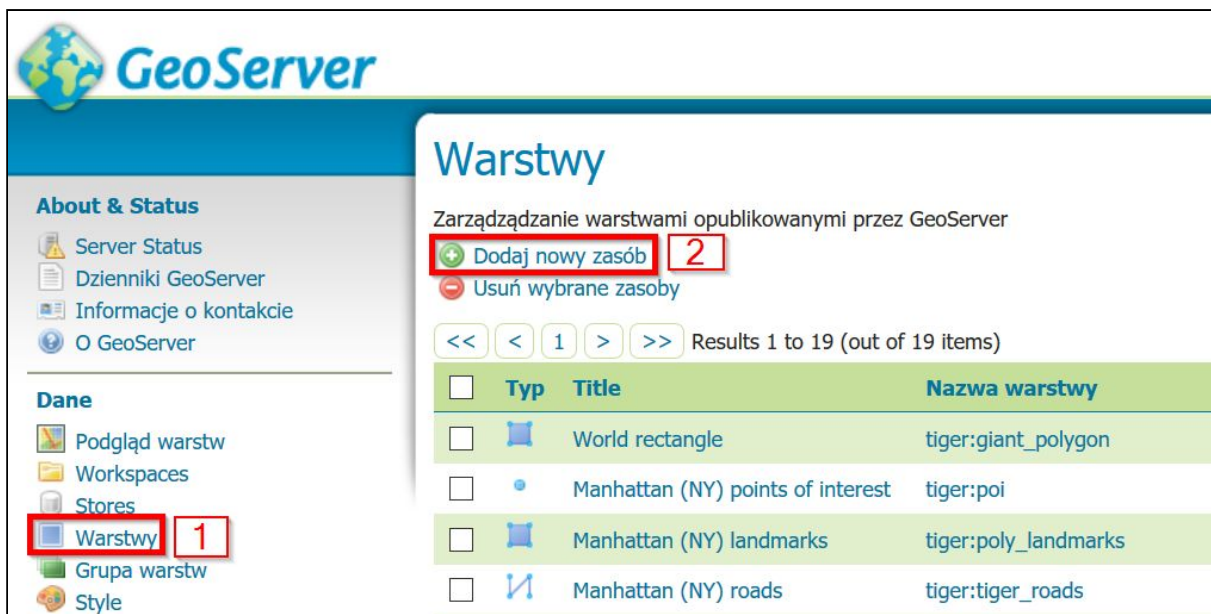
Ćwiczenie

Treść zadania

Dodaj cztery warstwy (*nmt*, *województwa*, *kondracki*, *miasta*) dla danych z utworzonych wcześniej magazynów.

Opis

W panelu administracyjnym *GeoServer* należy w sekcji *Dane* wybrać *Warstwy*. Będąc na stronie do zarządzania warstwami klikamy przycisk *Dodaj nowy zasób*.



The screenshot shows the GeoServer web interface. On the left sidebar, under the 'Dane' section, the 'Warstwy' menu item is highlighted with a red box and a '1' in a red box. In the main content area, the 'Warstwy' page is displayed. At the top, there is a 'Dodaj nowy zasób' button with a green plus icon, highlighted with a red box and a '2' in a red box. Below it is a 'Usuń wybrane zasoby' button with a red minus icon. A pagination bar shows 'Results 1 to 19 (out of 19 items)'. Below the pagination bar is a table with the following columns: 'Typ', 'Title', and 'Nazwa warstwy'. The table contains five rows of data:

Typ	Title	Nazwa warstwy
<input type="checkbox"/>	World rectangle	tiger:giant_polygon
<input type="checkbox"/>	Manhattan (NY) points of interest	tiger:poi
<input type="checkbox"/>	Manhattan (NY) landmarks	tiger:poly_landmarks
<input type="checkbox"/>	Manhattan (NY) roads	tiger:tiger_roads

Na kolejnej stronie należy wskazać magazyn, z którego będą pobierane informacje przestrzenne. W pierwszej kolejności dodamy numeryczny model terenu, więc wskaźmy

pozycję `webgis:nmt_store`. Pojawi się lista dostępnych warstw ze wskazanego źródła danych. Aby opublikować dane należy kliknąć opcję *Publish* przy wybranej warstwie.

New Layer

Add a new layer

Add layer from `webgis:nmt_store`

On stores you can also create a new coverage view by merging different coverages as a multibands coverage. [Configure new Coverage view ...](#)

Tu znajduje się lista zasobów w składnicy; 'nmt_store'. Naciśnij na warstwie, którą chcesz konfigurować

<< < 1 > >> Results 0 to 0 (out of 0 items)

Published	Layer name	Action
	dem84	Publish

<< < 1 > >> Results 0 to 0 (out of 0 items)

Na nowej stronie należy podać informacje o warstwie, które będą publikowane jako jej metadane. W części *Podstawowe informacje o zasobie* należy podać nazwę, która musi być unikalna w danym obszarze roboczym. Dla naszego rastra wpisujemy `nmt`. W polu *Tytuł* podajemy nazwę wyświetlaną użytkownikowi, może to być dowolny tekst np. *Numeryczny model terenu Polski*. Kolejnym ważnym krokiem jest sprawdzenie czy w sekcji *Granice* są wypełnione pola określające granice danych przestrzennych. Jeśli będą puste należy kliknąć opcje *Obliczyć na podstawie danych* oraz *Oblicz na podstawie natywnych granic*. Pozostałe elementy można pozostawić niezmienione.

Edit Layer

Edit layer data and publishing

webgis:dem84

Configure the resource and publishing information for the current layer

Dane Publishing Dimensions Buforowanie Kafelków Security

Edit Layer

Podstawowe informacje o zasobie

Nazwa
nmt

Włączone
 Advertised

Tytuł
Numeryczny model terenu Polski

Abstract

Granice

Natywna granica

Min X	Min Y	Max X	Max Y
14,1208333333	48,9958333333	24,1708333333000	54,8458333333

[Obliczyć na podstawie danych](#)
Compute from SRS bounds

Szerokość/Długość geograficzna granicy

Min X	Min Y	Max X	Max Y
14,1208333333	48,9958333333	24,1708333333000	54,8458333333

[Oblicz na podstawie natywnych granic](#)

Po wciśnięciu **Save** zostaniemy przekierowani na listę, na której powinna pojawić się utworzona warstwa.

Warstwy

Zarządzanie warstwami opublikowanymi przez GeoServer

+ Dodaj nowy zasób
- Usuń wybrane zasoby

<< < 1 > >> Results 1 to 20 (out of 20 items)
 🔍 Search

<input type="checkbox"/>	Typ	Title	Nazwa warstwy	Magazyn	Włączony?	Natywna SRS
<input type="checkbox"/>		Numeryczny model terenu Polski	webgis:nmt	nmt_store	✓	EPSG:4326
<input type="checkbox"/>		World rectangle	tiger:giant_polygon	nyc	✓	EPSG:4326
<input type="checkbox"/>		Manhattan (NY) points of interest	tiger:poi	nyc	✓	EPSG:4326
<input type="checkbox"/>		Manhattan (NY) landmarks	tiger:poly_landmarks	nyc	✓	EPSG:4326

Następnie dodajmy województwa z pliku ESRI Shapefile. Klikamy **Dodaj nowy zasób** i z listy wskazujemy `webgis:województwa_store`. Klikamy **Publish** przy warstwie `województwa`.

New Layer

Add a new layer

Add layer from webgis:województwa_store

You can create a new feature type by manually configuring the attribute names and types. [Create new feature type...](#)

Tu znajduje się lista zasobów w składnicy; 'województwa_store'. Naciśnij na warstwie, którą chcesz konfigurować

<< < 1 > >> Results 0 to 0 (out of 0 items)
 🔍 Search

Published	Layer name	Action
	województwa	Publish

<< < 1 > >> Results 0 to 0 (out of 0 items)
 🔍 Search

Ustawiamy te same opcje co w przypadku modelu terenu, jako nazwę zostawmy `województwa`, a tytuł `Województwa`.

Edit Layer

Edit layer data and publishing

webgis:województwa

Configure the resource and publishing information for the current layer

Dane **Publishing** **Dimensions** **Buforowanie Kafelków** **Security**

Edit Layer

Podstawowe informacje o zasobie

Nazwa
województwa

Włączone
 Advertised

Tytuł
Województwa

Abstract

Granice

Natywna granica

Min X	Min Y	Max X	Max Y
171 677,55013461	133 223,72223794	861 895,74660345	774 923,75012481

[Obliczyć na podstawie danych](#)
Compute from SRS bounds

Szerokość/Długość geograficzna granicy

Min X	Min Y	Max X	Max Y
13,8977816707890	48,9604161674784	24,6212961742371	54,8375661183030

[Oblicz na podstawie natywnych granic](#)

Klikamy **Save** i sprawdzamy czy nowa warstwa pojawiła się na liście.

Warstwy

Zarządzanie warstwami opublikowanymi przez GeoServer

[+ Dodaj nowy zasób](#)
[- Usuń wybrane zasoby](#)

<< < 1 > >> Results 1 to 21 (out of 21 items)

<input type="checkbox"/>	Typ	Title	Nazwa warstwy	Magazyn
<input type="checkbox"/>		Numeryczny model terenu Polski	webgis:nmt	nmt_store
<input type="checkbox"/>		Województwa	webgis:województwa	województwa_store
<input type="checkbox"/>		World rectangle	tiger:giant_polygon	nyc

Kolejne dwie warstwy przechowywane są w bazie *webgis*. Tak więc po kliknięciu *Dodaj nowy zasób* z listy magazynów wybieramy `webgis:postgis_store`. Najpierw dodamy warstwę z miastami, tak więc klikamy *Publish* przy tej pozycji.

New Layer

Add a new layer

Add layer from

You can create a new feature type by manually configuring the attribute names and types. [Create new feature type...](#)

On databases you can also create a new feature type by configuring a native SQL statement. [Configure new SQL view...](#)

Tu znajduje się lista zasobów w składnicy; 'postgis_store'. Naciśnij na warstwie, którą chcesz konfigurować

<< < 1 > >> Results 0 to 0 (out of 0 items)

Published	Layer name	Action
	kondracki	Publish
	miasta	Publish

<< < 1 > >> Results 0 to 0 (out of 0 items)

Jako nazwę pozostawiamy miasta, tytuł zapiszmy jako Miasta. Aktualizujemy również granice.

Edit Layer

Edit layer data and publishing

webgis:miasta

Configure the resource and publishing information for the current layer

Dane Publishing Dimensions Buforowanie Kafelków Security

Edit Layer

Podstawowe informacje o zasobie

Nazwa

Włączone
 Advertised

Tytuł

Abstract

Granice

Natywna granica

Min X	Min Y	Max X	Max Y
14,6141529083252	50,0335578918457	23,1588306427002	54,3593826293945

[Obliczyć na podstawie danych](#)
Compute from SRS bounds

Szerokość/Długość geograficzna granicy

Min X	Min Y	Max X	Max Y
14,6141529083252	50,0335578918457	23,1588306427002	54,3593826293945

[Oblicz na podstawie natywnych granic](#)

Klikamy Save i analogicznie dodajemy warstwę z tabeli kondracki, podając jako tytuł Podział fizjograficzny Polski.

Edit Layer

Edit layer data and publishing

webgis:kondracki

Configure the resource and publishing information for the current layer

Dane **Publishing** **Dimensions** **Buforowanie Kafelków** **Security**

Edit Layer

Podstawowe informacje o zasobie

Nazwa
kondracki

Włączone
 Advertised

Tytuł
Podział fizjograficzny Polski

Abstract

Granice

Natywna granica

Min X	Min Y	Max X	Max Y
172 892,703125	133 096,28125	861 029,125	775 242,25

[Obliczyć na podstawie danych](#)
Compute from SRS bounds

Szerokość/Długość geograficzna granicy

Min X	Min Y	Max X	Max Y
13,9162248100597	48,9597801978142	24,6083011264488	54,8404233234157

[Oblicz na podstawie natywnych granic](#)

Na liście warstw powinny pojawić się dwie kolejne pozycje.

Warstwy

Zarządzanie warstwami opublikowanymi przez GeoServer

Dodaj nowy zasób
 Usuń wybrane zasoby

<< < 1 > >> Results 1 to 23 (out of 23 items)





<input type="checkbox"/>	Typ	Title	Nazwa warstwy	Magazyn	Włączony?	Natywna SRS
<input type="checkbox"/>		Podział fizjograficzny Polski	webgis:kondracki	postgis_store		EPSG:2180
<input type="checkbox"/>		Miasta	webgis:miasta	postgis_store		EPSG:4326
<input type="checkbox"/>		Numeryczny model terenu Polski	webgis:nmt	nmt_store		EPSG:4326
<input type="checkbox"/>		Województwa	webgis:województwa	województwa_store		EPSG:2180
<input type="checkbox"/>		World rectangle	tiger:giant_polygon	nyc		EPSG:4326
<input type="checkbox"/>		Manhattan (NY) points of interest	tiger:poi	nyc		EPSG:4326

Podgląd warstw

Podgląd warstw

Lista wszystkich skonfigurowanych warstw w GeoServer i podgląd0 każdego w różnych formatach.

<< < 1 > >> Results 1 to 22 (out of 22 items) Search

Typ	Tytuł	Nazwa	Typowe formaty	Wszystkie formaty
	World rectangle	tiger:giant_polygon	OpenLayers GML KML	Wybierz jedną
	Manhattan (NY) points of interest	tiger:poi	OpenLayers GML KML	Wybierz jedną
	Manhattan (NY) landmarks	tiger:poly_landmarks	OpenLayers GML KML	Wybierz jedną
	Manhattan (NY) roads	tiger:tiger_roads	OpenLayers GML KML	Wybierz jedną

Dodane warstwy można podejrzeć w panelu administracyjnym w zakładce *Podgląd warstw*. W zależności od tego z jakim typem warstwy mamy do czynienia (raster, wektor) mogą być widoczne różne formaty do wyboru, które pozwalają pobrać dane. Szczególną opcją jest *OpenLayers* pozwalająca na bezpośrednio podgląd warstwy w przeglądarce w formie interaktywnej mapy. Po jej wybraniu otworzy się nowe okno przeglądarki i zostanie wyświetlona prosta mapka z danymi.

Na liście *Wszystkie formaty* dostępne są wszystkie wspierane przez Geoserver formaty danych, które służą do ich publikacji. W grupie *WMS* są dostępne formaty rastrowe, a w *WFS* wektorowe. Po wskazaniu opcji dane zostaną pobrane w wybranym formacie. Można je zapisać i podejrzeć w aplikacji np. *QGIS*.

Publikacja danych

Geoserver udostępnia dane w formie usług m.in. *WMS*, *WMTS* i *WFS*. Aby z nich skorzystać należy znać ich adres, który jest podawany w aplikacji klienckiej. Każda z usług posiada zdefiniowany zestaw parametrów, które są dołączane do adresu podczas komunikacji z serwerem. Dzięki temu klient może pobrać metadane danej usługi (tzw. *Capabilities*) zawierające m.in. listę warstw, dostępne układy współrzędnych, formaty i style, a także otrzymywać dane przestrzenne do wyświetlenia na mapie.

Aby znaleźć adres dla poszczególnych usług można skorzystać z panelu administracyjnego. Na stronie głównej po prawej stronie jest lista dostępnych usług. Można kliknąć na interesującą nas wersję i zostaniemy pobrany XML zawierający metadane usługi. W przypadku *WMS* może to wyglądać w ten sposób:

<http://127.0.0.1:8080/geoserver/ows?service=wms&version=1.3.0&request=GetCapabilities>

To co nas interesuje to część przed znakiem "?", czyli <http://127.0.0.1:8080/geoserver/ows>.

W przypadku *WMTS* adres może wyglądać tak:

<http://127.0.0.1:8080/geoserver/gwc/service/wmts?REQUEST=GetCapabilities> - tu również interesuje nas część przed znakiem "?".

Adresy w tej formie wyświetlają warstwy ze wszystkich obszarów roboczych, chyba, że dany obszar został oznaczony jako izolowany (*Isolated workspace*). Istnieje jednak możliwość zmodyfikowania adresu tak, aby pobierane były dane tylko dla wybranego obszaru. W tym

celu należy dopisać jego nazwę po części `/geoserver/`. Przykładowo, chcąc wyświetlić warstwę z obszaru roboczego `webgis` dla usługi `WMS` należy podać adres <http://127.0.0.1:8080/geoserver/webgis/ows>, a dla `WMTS` <http://127.0.0.1:8080/geoserver/webgis/gwc/service/wmts>. Dzięki temu można udostępniać różne adresy z jednej instancji `Geoserver`, zawierające osobne zestawy danych. W dalszej części będziemy korzystać z adresów zawierających nazwę obszaru roboczego `webgis`, ponieważ interesują nas tylko dane z niego.

Wczytanie danych do QGIS

Dane bezpośrednio z `Geoserver` można wczytać do dowolnej aplikacji, która obsługuje usługi `OGC`, w tym do `QGIS`. Służą do tego odpowiednie zakładki w oknie *Zarządzanie źródłami danych*.

WMS/WMTS

Aby dodać nową usługę rastrową należy przejść do zakładki `WMS/WMTS`. Następnie klikamy przycisk *Nowa*.

Utwórz nowe połączenie WMS/WMTS

Szczegóły połączenia

Nazwa

URL

Uwierzytelnianie

Konfiguracja Bez zabezpieczeń

Wybierz lub utwórz konfigurację uwierzytelniania

Bez uwierzytelniania

Konfiguracja przechowuje zaszyfrowane dane w bazie danych uwierzytelniania QGIS.

Opcje WMS/WMTS

Referer

Tryb DPI

Ignoruj GetMap/GetTile URI podany w Capabilities

Ignoruj GetFeatureInfo URI podany w Capabilities

Ignoruj orientację osi (WMS 1.3/WMTS)

Ignoruj zgłoszone zakresy warstw

Odwróć orientację osi

Wyglądź przekształcenie bitmap

W nowym oknie należy podać nazwę i adres URL do `Geoserver`. Dla `WMS` jest to <http://127.0.0.1:8080/geoserver/webgis/ows>, a dla usługi `WMTS` <http://127.0.0.1:8080/geoserver/webgis/gwc/service/wmts>. Po zapisaniu należy kliknąć przycisk *Połącz*. W przypadku `WMS` zostanie wyświetlona lista dostępnych warstw, można wskazać jedną lub więcej warstw. Można dodatkowo zdefiniować układ współrzędnych

klikając przycisk *Zmień*. Dla *WMTS* warstwy zostaną wylistowane w zakładce *Kafle*. Jedna warstwa może być wyświetlona kilkakrotnie, osobno dla różnych formatów i układów współrzędnych. Możliwy jest wybór pojedynczego elementu. Po wskazaniu warstwy należy kliknąć przycisk *Dodaj*. Warstwa powinna wczytać pojawić się w oknie *QGIS*.

WFS

Dodawanie danych wektorowych odbywa się w zakładce *WFS* i jest niemal identyczne jak w przypadku usługi *WMS*.

Utwórz nowe połączenie WFS

Szczegóły połączenia

Nazwa

URL

Uwierzytelnianie

Konfiguracja Bez zabezpieczeń

Wybierz lub utwórz konfigurację uwierzytelniania

Bez uwierzytelniania

Konfiguracja przechowuje zaszyfrowane dane w bazie danych uwierzytelniania QGIS.

Opcje WFS

Wersja Maksimum Rozpoznaj

Maks. liczba obiektów

Włącz stronicowanie obiektów

Rozmiar strony

Ignoruj orientację osi (WFS 1.1/WFS 2.0)

Odwróć orientację osi

OK Anuluj Pomoc

Po kliknięciu przycisku *Nowy* pojawi się okno gdzie wpisujemy nazwę, adres <http://127.0.0.1:8080/geoserver/ows> i zatwierdzamy przyciskiem *OK*. Po kliknięciu *Połącz* pojawi się lista dostępnych warstw, można wskazać jedną lub więcej pozycji oraz wybrać układ współrzędnych dla danych. Aby wczytać warstwę do *QGIS* należy kliknąć przycisk *Dodaj*.

Stylizacja

Style

Zarządzaj stylami opublikowanymi przez GeoServer

Dodaj nowy styl

Usunięto zaznaczony styl(e)

<< < 1 > >> Results 1 to 21 (out of 21 items)

<input type="checkbox"/>	Nazwa	Obszar roboczy
<input type="checkbox"/>	burg	
<input type="checkbox"/>	capitals	
<input type="checkbox"/>	cite_lakes	
<input type="checkbox"/>	dem	
<input type="checkbox"/>	generic	
<input type="checkbox"/>	giant_polygon	

Każda warstwa może mieć przypisany jeden lub więcej stylów, które są używane do renderowania danych. Style są wykorzystywane przez *Geoserver* do renderowania obrazów przedstawiających dane warstw np. w usłudze *WMS*. Domyślnie obsługiwany format *Styled Layer Descriptor* (SLD), ale poprzez rozszerzenia możliwe jest dodanie innych formatów m.in. CSS.

Format SLD jest standardem *OGC* do opisywania wyglądu warstw, zarówno wektorowych jak i rastrowych. Jest on oparty o schemat XML, poszczególne elementy są opisywane znacznikami z atrybutami, które można zagnieżdżać tworząc strukturę hierarchiczną. Specyfikacja tego formatu jest szczegółowo omówiona w dokumentacji *Geoserver*: <https://docs.geoserver.org/latest/en/user/styling/sld/index.html>.

Aby dodać nowy styl należy kliknąć *Dodaj nowy styl*.

Nowy styl

Wprowadź nową definicję SLD, użyj istniejącej jako wzoru lub załaduj przygotowany styl z systemu plików. Edytor dostarcza funkcjonalności podświetlania składni i może być powiększony do pełnego ekranu. Naciśnij przycisk "waliduj", aby sprawdzić czy styl jest prawidłowym plikiem SLD.

Data

Style Data

Nazwa

Obszar roboczy

Format
SLD

Style Content

Generate a default style
Wybierz [Generate ...](#)

Kopiuj z istniejącego stylu
Wybierz [Kopia ...](#)

Upload a style file
[Przeglądaj...](#) Nie wybrano pliku. [Załaduj ...](#)

Legend

[Add legend](#)

[Preview legend](#)

1

[Waliduj](#) [Apply](#) [Wyślij](#) [Cancel](#)

W części *Style data* należy podać nazwę stylu, obszar roboczy oraz format. Na dole strony widoczny jest prosty edytor tekstowy, w którym można definiować styl. Aby nie pisać stylu od zera można dodać treść do edytora na kilka sposobów:

- **Generate a default style** - należy wskazać typ danych i kliknąć *Generate*, zostanie wpisany prosty styl SLD z szablonu,
- **Kopiuj z istniejącego stylu** - można wybrać istniejący już styl i skopiować do edytora,
- **Upload a style file** - wybranie pliku ze stylem z dysku, zawartość pliku można też wkleić ręcznie.

Na samym dole strony widoczne są przyciski opcji:

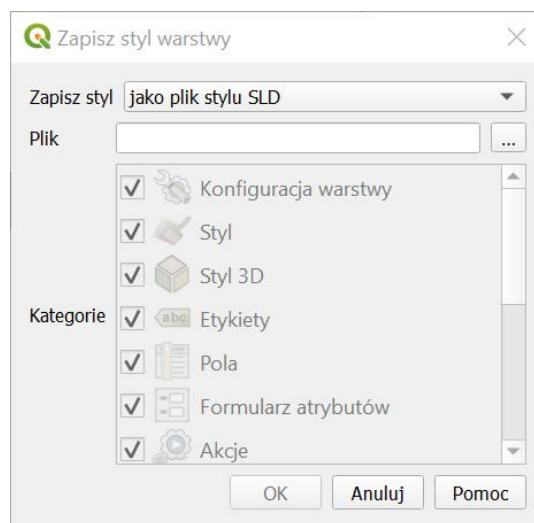
- **Waliduj** - sprawdza poprawność stylu wg specyfikacji SLD, jeśli zostaną znalezione błędy to zostaną one wyświetlone, nie można zapisać stylu zawierającego błędy,
- **Apply** - zapisanie zmian,
- **Wyślij** - zapisanie zmian i wrócenie do listy stylów,
- **Cancel** - powrót do listy stylów bez zapisywania zmian.

Klikając link *Preview legend* możliwe jest podejrzanie stylizacji w formie graficznej reprezentacji.

Symbolizacja danych w QGIS

Ręczne tworzenie stylów, nawet wykorzystując istniejące szablony i przykłady, jest zadaniem pracochłonnym. Szczególnie jeśli chcemy uwzględnić np. wartości atrybutów lub etykietowanie. Na szczęście istnieją narzędzia, które ułatwiają przygotowywanie wizualizacji. W naszym przypadku skorzystamy z *QGIS*, który umożliwia eksport stylu warstwy do formatu SLD. Należy jednak zaznaczyć, że bardziej zaawansowane funkcjonalności (np. stylizacja oparta o dane) mogą nie być wspierane przez format SLD, w związku z czym ostateczny styl może nieco się różnić od tego co jest widoczne w *QGIS*.

Aby wyeksportować styl należy wejść we właściwości warstwy. Na samym dole widoczny jest przycisk *Styl*, po jego rozwinięciu z menu wybieramy opcję *Zapisz styl*. W nowym oknie z listy *Zapisz styl* wybieramy opcję *jako plik stylu SLD* i wskazujemy lokalizację pliku wyjściowego. Klikamy *OK*, plik zostanie utworzony i można do załadować do aplikacji *Geoserver*.



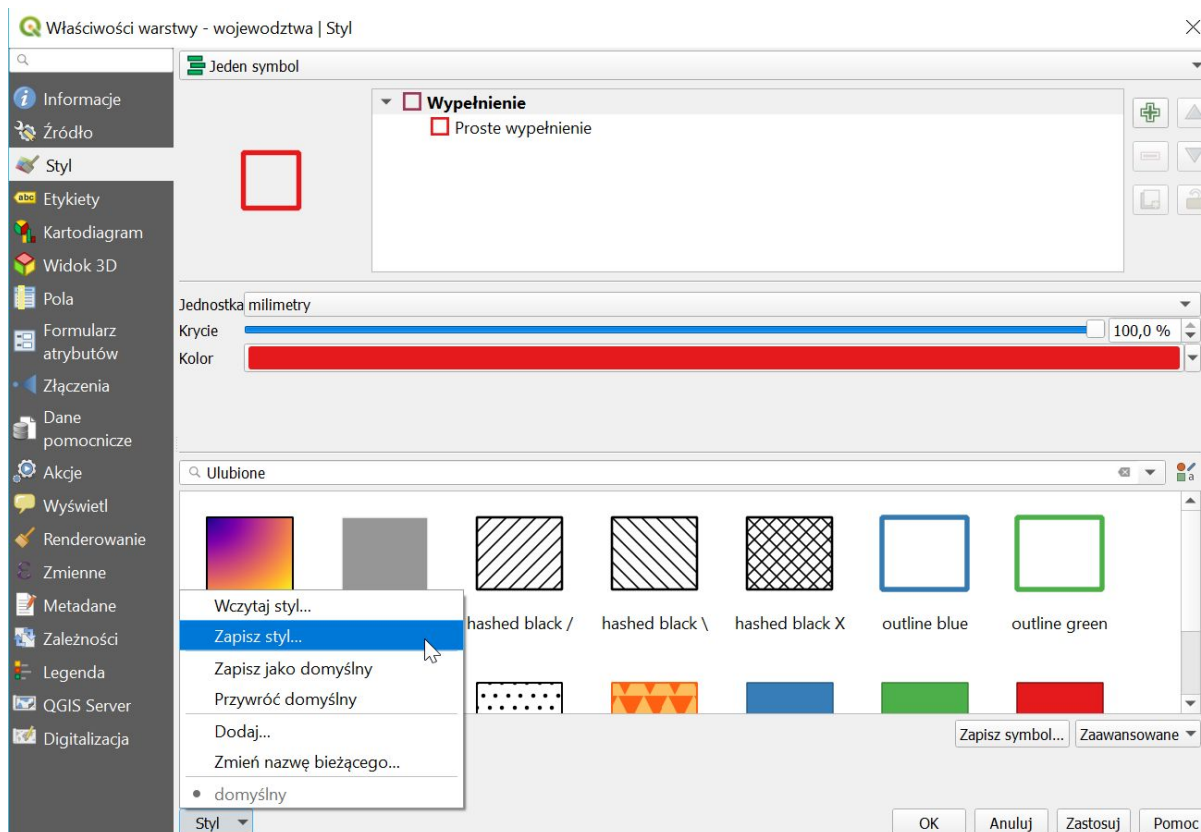
Ćwiczenie

Treść zadania

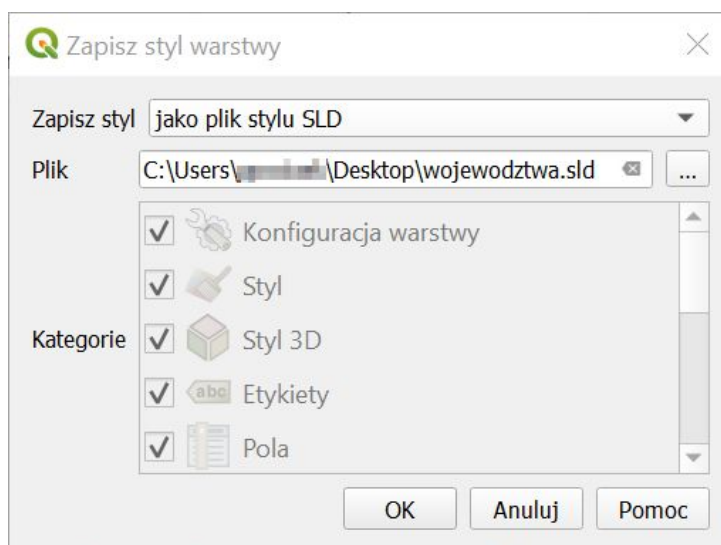
Dodaj style do czterech testowych warstw (*nmt*, *województwa*, *kondracki*, *miasta*). Wszystkie style należy wyeksportować z *QGIS*.

Opis

W pierwszej kolejności wyeksportujemy style z *QGIS* do plików w formacie SLD. W tym celu należy kolejno wejść do właściwości każdej z warstw, na dole okna kliknąć przycisk *Styl* i wybrać opcję *Zapisz styl*.



W nowym oknie wybieramy opcję zapisu *jako plik stylu SLD* i wskazujemy lokalizację na dysku. Dla uproszczenia nazwijmy pliki tak jak nazwy warstw w Geoserver tj. nmt.sld, wojewodztwa.sld, kondracki.sld i miasta.sld.



Mając gotowe style można dodać je w Geoserver. W panelu administracyjnym należy w sekcji *Dane* wybrać *Style*. Będąc na stronie do zarządzania stylami klikamy przycisk *Dodaj nowy styl*.

The screenshot shows the GeoServer web interface. On the left sidebar, the 'Style' menu item is highlighted with a red box and the number '1'. In the main content area, the 'Style' management page is displayed. At the top of this page, the 'Dodaj nowy styl' button is highlighted with a red box and the number '2'. Below the buttons, there is a table of styles with columns 'Nazwa' and 'Obszar roboczy'. The styles listed are 'burg', 'capitals', 'cite_lakes', and 'dem'.

Na stronie dodawania należy podać nazwę stylu i wybrać obszar roboczy *webgis*. Zaczniemy od warstwy województw, więc nazwę zapiszmy jako *województwa_style*. Następnie klikamy przycisk *Przeglądaj* i wskazujemy plik ze stylem *województwa.sld* i *Załaduj* w celu wczytania danych z pliku do pola tekstowego. W celu sprawdzenia czy styl jest poprawny należy kliknąć przycisk *Waliduj* oraz można wygenerować podgląd przyciskiem *Preview legend*.

No validation errors.

Edytor Stylu

Edytuj aktualny styl SLD. Edytor dostarcza funkcje podświetlenia składni i może być powiększony do pełnego ekranu. Naciśnij przycisk "waliduj" aby sprawdzić czy styl jest poprawnym dokumentem SLD.

Data **Publishing** **Layer Preview** **Layer Attributes**

Style Data

Nazwa

Obszar roboczy

Format
 Format only editable for new styles

Style Content

Generate a default style
 [Generate ...](#)

Kopiuż z istniejącego stylu

Kopia ...

Upload a style file
 Nie wybrano pliku. [Załaduj ...](#)

Legend

Legend

[Preview legend](#)

Single symbol

Font Height

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <StyledLayerDescriptor xmlns="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="1.1.0" xmlns:se="http://www.opengis.net/se"
  xsi:schemaLocation="http://www.opengis.net/sld http://schemas.opengis.net/sld/1.1.0
  /StyledLayerDescriptor.xsd">
3   <NamedLayer>
4     <se:Name>województwa</se:Name>
5     <UserStyle>
6       <se:Name>województwa</se:Name>
7       <se:FeatureTypeStyle>
8         <se:Rule>
9           <se:Name>Single symbol</se:Name>
10          <se:PolygonSymbolizer>
11            <se:Stroke>
12              <se:SvgParameter name="stroke">#e41a1c</se:SvgParameter>
13              <se:SvgParameter name="stroke-width">3</se:SvgParameter>
14              <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
15            </se:Stroke>
16          </se:PolygonSymbolizer>
17        </se:Rule>
18      </se:FeatureTypeStyle>
19    </UserStyle>
20  </NamedLayer>
21 </StyledLayerDescriptor>

```

No validation errors.

Na koniec klikamy **Wyślij** i zostaniemy przekierowani na listę stylów, na której powinna być widoczna nowa pozycja.

Style

Zarządzaj stylami opublikowanymi przez GeoServer

Results 1 to 1 (out of 1 matches from 25 items)

<input type="checkbox"/>	Nazwa	Obszar roboczy
<input type="checkbox"/>	województwa_style	webgis

Results 1 to 1 (out of 1 matches from 25 items)

Analogicznie dodajemy style dla pozostałych warstw podając ich nazwy: `nmt_style` (dla modelu terenu), `miasta_style` i `kondracki_style` (dla danych z bazy `webgis`).

Style

Zarządzaj stylami opublikowanymi przez GeoServer

Results 1 to 25 (out of 25 items)

<input type="checkbox"/>	Nazwa	Obszar roboczy
<input type="checkbox"/>	kondracki_style	webgis
<input type="checkbox"/>	miasta_style	webgis
<input type="checkbox"/>	nmt_style	webgis
<input type="checkbox"/>	województwa_style	webgis

Przypisanie stylu do warstwy

Aby przypisać styl do warstwy należy w zakładce *Style* kliknąć nazwę docelowej warstwy na liście i przejść do zakładki *Publishing*. W części *WMS Settings/Layer Settings* można zdefiniować style warstwy. *Domyślny styl* pozwala określić styl, w jakim będzie widoczna warstwa np. jeśli nie podano innego stylu. Dodatkowo poniżej można dodać inne style, w których dana warstwa może być wyświetlana. Aby to zrobić trzeba przenieść elementy z listy *Dostępne style* do *Wybrane style* zaznaczając wybrane style i klikając przycisk \Rightarrow . Wyboru stylu dokonuje aplikacja kliencka i przekazuje tę informację w adresie podczas pobierania danych.

Ćwiczenie

Treść zadania

Przypisz dodane wcześniej style odpowiednim warstwom ćwiczeniowym.

Opis

Aby przypisać styl należy wejść w panelu administracyjnym do listy warstw i kliknąć nazwę warstwy. W pierwszej kolejności przypiszemy styl województwom.

The screenshot shows the GeoServer 'Warstwy' (Layers) page. On the left sidebar, the 'Warstwy' menu item is highlighted with a red box and the number '1'. The main content area displays a table of layers with the following data:

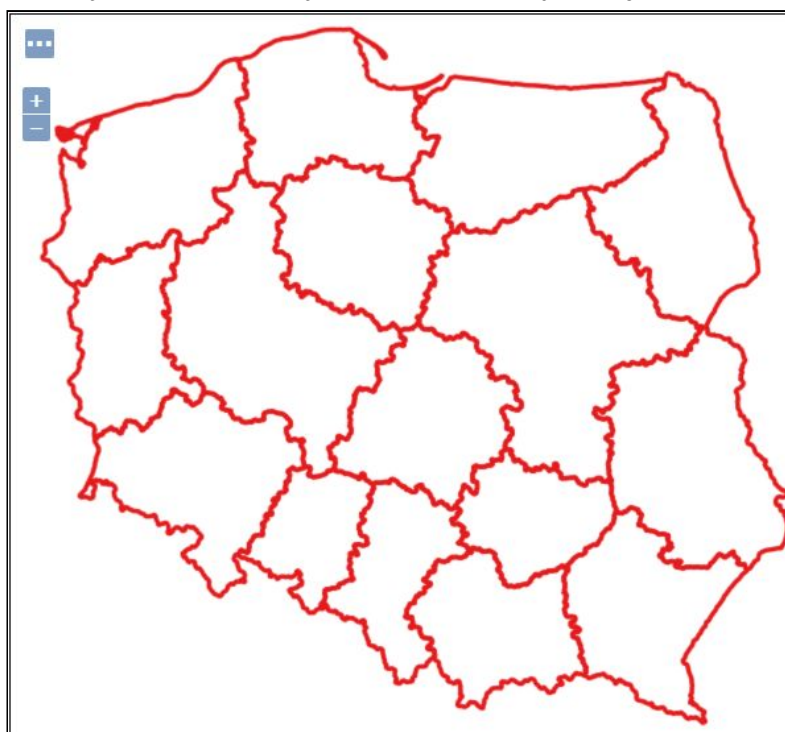
Typ	Title	Nazwa warstwy	Magazyn
<input type="checkbox"/>	Podział fizjograficzny Polski	webgis:kondracki	postgis_store
<input type="checkbox"/>	Miasta	webgis:miasta	postgis_store
<input type="checkbox"/>	Numeryczny model terenu Polski	webgis:nmt	nmt_store
<input type="checkbox"/>	Województwa	webgis:województwa	województwa_store

The 'Województwa' row is highlighted in red, and the number '2' is placed next to the 'webgis:województwa' value.

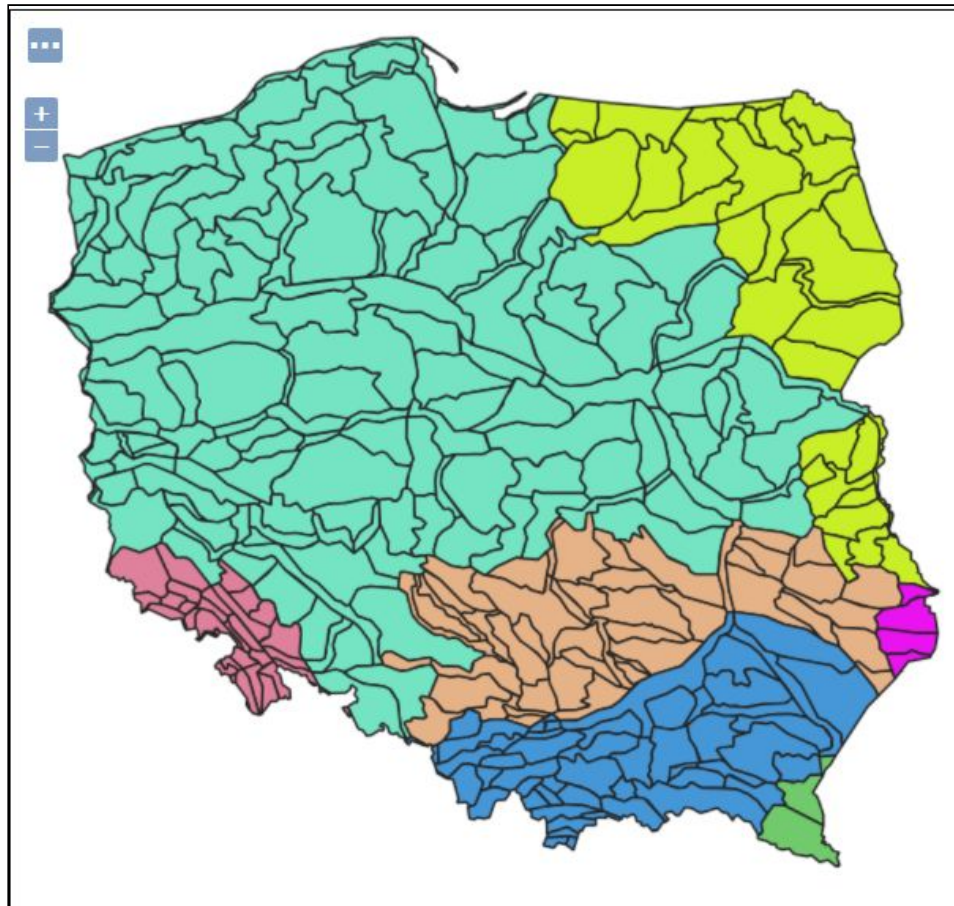
We właściwościach warstwy przechodzimy do zakładki *Publishing* i przewijamy do sekcji *Layer Settings*. Następnie z listy *Domyślny styl* należy wybrać zdefiniowany styl, w przypadku województw jest to `webgis:województwa_style`.

The screenshot shows the 'WMS Settings Layer Settings' dialog box. The 'Domyślny styl' (Default style) dropdown menu is set to 'webgis:województwa_style'. There are also checkboxes for 'Queryable' (checked) and 'Nieprzezroczysty' (unchecked).

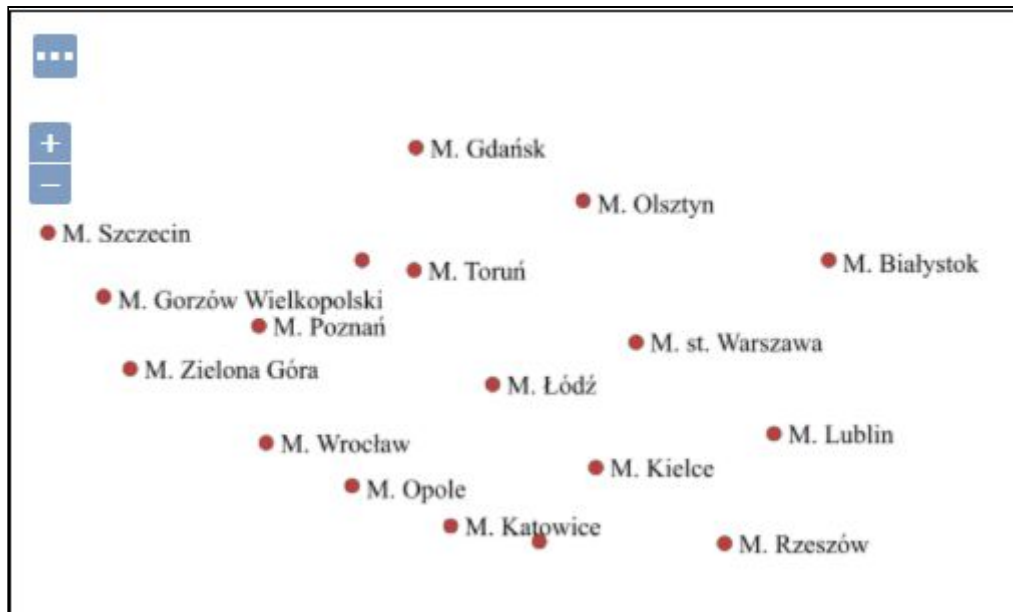
Następnie klikamy *Save* na dole strony. Aby zweryfikować czy styl został poprawnie dodany należy wejść w *Podgląd warstw* i kliknąć *OpenLayers* przy danej warstwie.



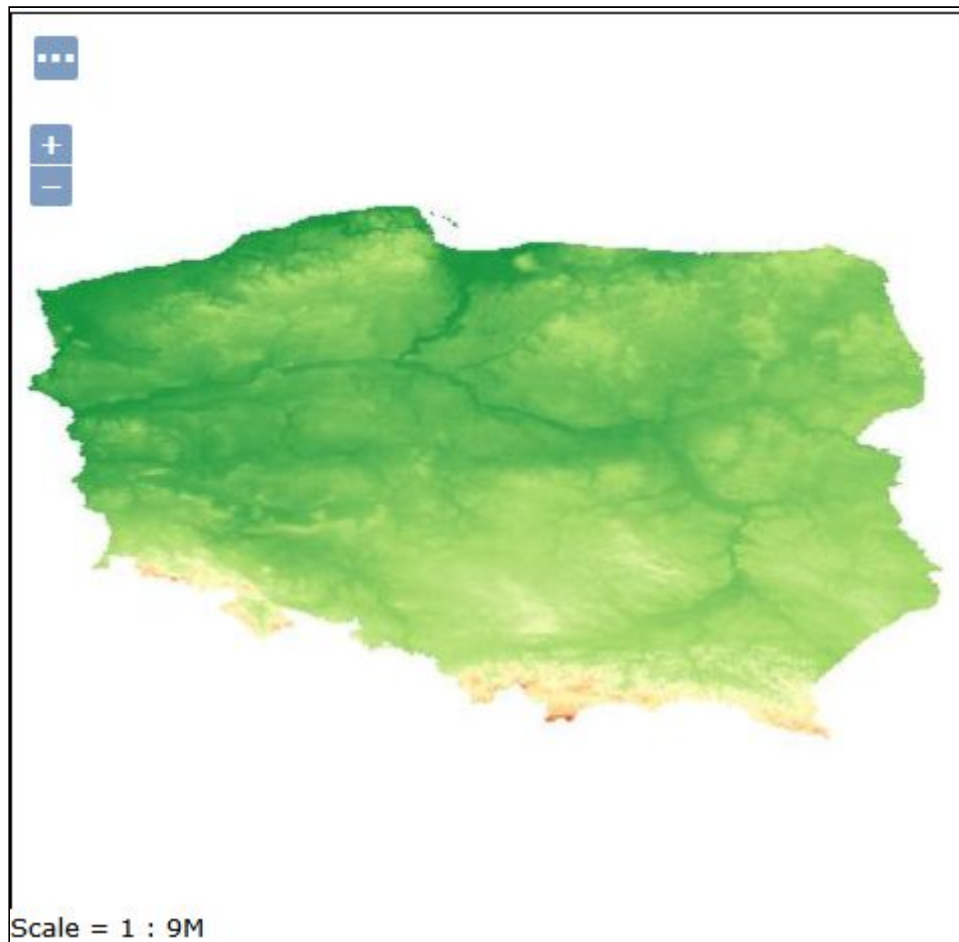
Analogicznie należy przypisać styl dla pozostałych warstw. Podział fizjograficzny Polski:



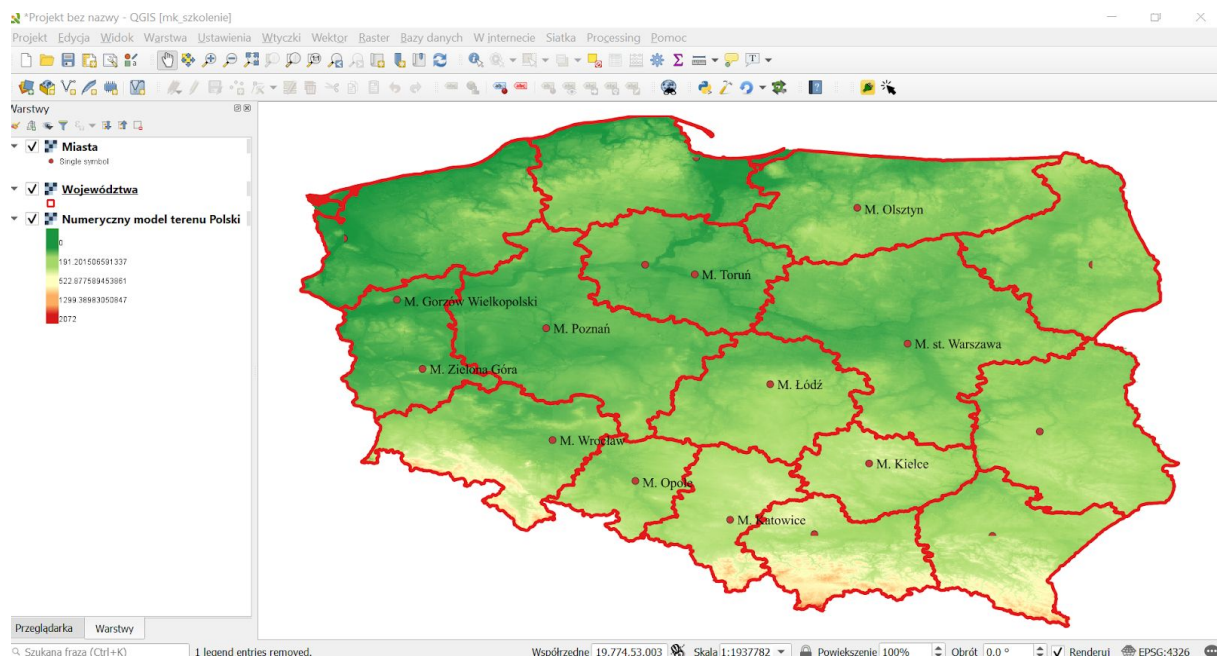
Numeryczny model terenu:



Miasta:



Nowe style będą również widoczne w QGIS przy korzystaniu z usług zwracających obraz np. WMS i WMTS.



Tworzenie serwisów internetowych

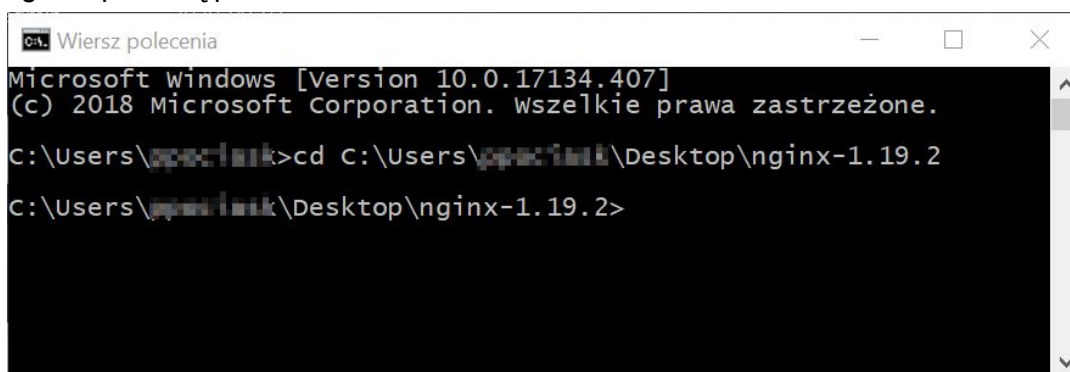
Serwer WWW

Apache i *nginx* to dwa najpopularniejsze serwery WWW na świecie. Łącznie obsługują (wg różnych szacunków) od 60 do 90% ruchu w Internecie. Służą do obsługi przychodzących żądań, potrafią rozdzielać ruch pomiędzy kilka niezależnych od siebie aplikacji webowych. Oba projekty są udostępniane na otwartej licencji i można je wykorzystywać do celów komercyjnych bez ograniczeń. Posiadają również możliwość instalowania rozszerzeń, które dodają nowe funkcjonalności. Większość serwerów jest obsługiwana przez systemy z rodziny *Linux*, jednak istnieją również wersje aplikacji na systemy *Windows*.

W ramach szkolenia wykorzystamy aplikację *nginx for Windows*, która jest opisana na stronie <http://nginx.org/en/docs/windows.html>. Po przejściu na stronę pobierania można wybrać jedną z dwóch wersji:

- **stable** - wersja stabilna, zalecana na serwery produkcyjne, trafiają do niej jedynie poprawki do błędów, bez nowych funkcjonalności,
- **mainline** - wersja aktualnie rozwijana, poza poprawkami błędów mogą do niej trafiać również nowe funkcjonalności.

Po pobraniu wersji na *Windows* należy rozpakować archiwum w dowolnym katalogu. Aplikację uruchamia się z systemowego wiersza poleceń. W *Windows* jest to aplikacja *cmd.exe*, którą można znaleźć w menu *Start*. Następnie należy przejść do utworzonego katalogu za pomocą polecenia `cd`.



```
ca. Wiersz polecenia
Microsoft windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\apoc1\>cd C:\Users\apoc1\Desktop\nginx-1.19.2
C:\Users\apoc1\Desktop\nginx-1.19.2>
```

Aby uruchomić *nginx* należy wpisać polecenie `start nginx`. Aby zweryfikować czy aplikacja uruchomiła się poprawnie i działa w tle należy wywołać polecenie:

```
tasklist /fi "imagename eq nginx.exe"
```

Powinna pojawić się krótka lista uruchomionych procesów *nginx*:

```
ca. Wiersz polecenia
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. Wszelkie prawa zastrzeżone.
C:\Users\ppoc\Desk>cd C:\Users\ppoc\Desk\Desktop\nginx-1.19.2
C:\Users\ppoc\Desk\Desktop\nginx-1.19.2>start nginx
C:\Users\ppoc\Desk\Desktop\nginx-1.19.2>tasklist /fi "imagename eq nginx.exe"
Image Name                PID Session Name        Session#    Mem Usage
=====
nginx.exe                 5224 Console             2           7 320 K
nginx.exe                 708 Console             2           7 600 K
C:\Users\ppoc\Desk\Desktop\nginx-1.19.2>
```

Do sterowania aplikacją nginx służą polecenia:

- `nginx -s quit` - zamknięcie aplikacji,
- `nginx -s reload` - restart aplikacji np. po zmianach w konfiguracji.

```
ca. Wiersz polecenia
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. Wszelkie prawa zastrzeżone.
C:\Users\ppoc\Desk>cd C:\Users\ppoc\Desk\Desktop\nginx-1.19.2
C:\Users\ppoc\Desk\Desktop\nginx-1.19.2>start nginx
C:\Users\ppoc\Desk\Desktop\nginx-1.19.2>tasklist /fi "imagename eq nginx.exe"
Image Name                PID Session Name        Session#    Mem Usage
=====
nginx.exe                 5224 Console             2           7 320 K
nginx.exe                 708 Console             2           7 600 K
C:\Users\ppoc\Desk\Desktop\nginx-1.19.2>nginx -s reload
C:\Users\ppoc\Desk\Desktop\nginx-1.19.2>nginx -s quit
C:\Users\ppoc\Desk\Desktop\nginx-1.19.2>tasklist /fi "imagename eq nginx.exe"
INFO: No tasks are running which match the specified criteria.
C:\Users\ppoc\Desk\Desktop\nginx-1.19.2>
```

Konfiguracja nginx

Głównym plik konfiguracyjny nginx znajduje się w folderze *conf* i nazywa się *nginx.conf*. Określa się w nim tzw. dyrektywy, które sterują zachowaniem aplikacji. Są dwa rodzaje dyrektyw:

- **proste** - jednolinijkowe, składa się ona z nazwy i wartości rozdzielonych spacją i zakończonych średnikiem,
- **blokowe** - składają się z nazwy, po której wewnątrz klamr { } definiowane są parametry danej dyrektywy.

W pliku *nginx.conf* znajduje się przykładowa konfiguracja, z której można skorzystać. Teksty rozpoczynające się od # to komentarze i są ignorowane podczas uruchamiania aplikacji.

Głównym elementem jest dyrektywa `http`, w której definiowane są konfiguracje dla obsługiwanych serwerów. W pliku powinna być jedna dyrektywa `http`.

Wewnątrz może znajdować się jedna lub więcej dyrektyw `server`. Pozwalają one określić tzw. wirtualne serwery posiadające osobną konfigurację. Każdy taki serwer powinien mieć zdefiniowany port, który nasłuchuje (dyrektywa `listen`) oraz nazwę określającą zazwyczaj domeny serwera (`server_name`). Wirtualny serwer może przetworzyć przychodzące żądanie, `nginx` decyduje na podstawie ich konfiguracji, który to zrobi. Dyrektywa `server` musi znajdować się wewnątrz `http`.

```
http {
    server {
        listen 80;
        server_name www.domena1.com domena1.com;
        ...
    }
    server {
        listen 8000;
        server_name www.domena2.com domena2.com;
        ...
    }
}
```

W powyższym przykładzie są dwie dyrektywy blokowe (`http` i dwie `server`) oraz dwie proste (`listen`, `server_name`). Zdefiniowane są dwa serwery wirtualne. Pierwszy nasłuchuje na porcie 80 (domyślny port dla HTTP) pierwszą domenę, żeby żądanie zostało przez niego przetworzone użytkownik musi wpisać w przeglądarkę adres: <http://www.domena1.com> lub <http://domena1.com>. Port 80 jest domyślny więc może zostać pominięty w adresie.

Drugi serwer obsługuje port 8000, żeby się do niego odwołać należy wpisać <http://www.domena2.com:8000> lub <http://domena2.com:8000>. W tym wypadku konieczne jest podanie numeru portu.

Ostatnią ważną wyrektywą blokową, którą poznamy jest `location`. Pozwala ona na podstawie adresu URL (a właściwie jego części po domenie) wykonać wskazaną operację. Może to być zwrócenie wskazanego pliku (statyczne serwowanie) lub przekazanie żądania do aplikacji webowej (*proxy*). Każdy `server` powinien mieć przynajmniej jedną dyrektywę `location`, może ich być jednak więcej. Reguły dla adresów podawane są zaraz po nazwie `location`.

```
http {
    server {
        server_name strona.pl

        location / {
            # reguła 1
            ...
        }
    }
}
```

```

    location /app1 {
        # reguła 2
        ...
    }

    location /app2 {
        # reguła 3
        ...
    }
}

```

W powyższym przykładzie dla domeny `strona.pl` zdefiniowane są 3 reguły. Pierwsza z nich określa ogólny przypadek, który obsłuży wszystkie adresy nie pasujące do innych reguł. Jeśli użytkownik wpisze w przeglądarce <http://strona.pl>, <http://strona.pl/podstrona> lub <http://strona.pl/plik.zip> to wszystkie te adresy zostaną obsłużone przez regułę 1. Natomiast jeśli użytkownik wprowadzi adres <http://strona.pl/app1> lub <http://strona.pl/app1/podstrona> to *nginx* prześle ich obsługę do reguły 2. Analogicznie adresy rozpoczynające się od <http://strona.pl/app2> trafią do reguły 3.

Dzięki takiemu zachowaniu z poziomu jednej aplikacji *nginx* możliwe jest obsługiwanie wielu aplikacji webowych. W definiowaniu reguł dla adresów można również wykorzystać wyrażenia regularne do zaawansowanego kierowania ruchem.

Pliki statyczne

Styczna obsługa plików polega na zwracaniu plików wskazanych w adresie URL. Są one przechowywane w katalogu, do którego ma dostęp *nginx*. Przykładowo adres <https://domena.com/index.html> zwraca dokument HTML o nazwie *index.html*, który jest przechowywany w zdefiniowanym folderze na dysku.

Aby obsługiwać pliki statycznie *nginx* musi wiedzieć w jakim katalogu znajdują się pliki. Określa się to za pomocą dyrektywy `root`. Ważne jest, aby *nginx* miał odpowiednie uprawnienia do podanego katalogu np. do odczytu zawartości plików.

```

location / {
    root html;
    index index.html;
}

```

W powyższym przykładzie *nginx* będzie wyszukiwał pliki w katalogu znajdującym się wewnątrz folderu aplikacji, w podkatalogu `html` np. <http://strona.pl/plik.zip>. Jeśli plik o podanej nazwie nie istnieje zostanie zwrócony błąd *404 Not Found*.

Widoczna dyrektywa `index` pozwala zwrócić podany plik (w tym przypadku `C:\data\www\index.html`) jeśli użytkownik wpisze adres bez podania pliku czyli <http://strona.pl>.

Ćwiczenie

Treść zadania

Utwórz na pulpicie nowy katalog *webgis*, a w nim nowy plik tekstowy o nazwie *index.html*. Następnie zmodyfikuj ustawienia *nginx*, tak aby główny adres / wskazywał na nowy folder. Po uruchomieniu *nginx* adres <http://127.0.0.1> powinien wskazywać na stronę z pliku *index.html*.

Zawartość pliku *index.html*:

```
<html lang="pl">
  <head>
    <meta charset="UTF-8">
    <title>WebGIS</title>
  </head>
  <body>
    <h2>Strona główna</h2>
  </body>
</html>
```

Opis

Na początku tworzymy katalog *webgis* na pulpicie i w nim plik tekstowy *index.html*. Plik należy wypełnić zgodnie z treścią zadania.

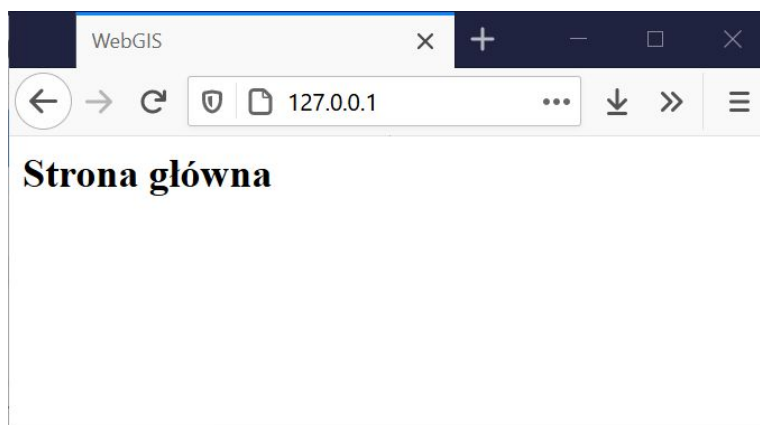
Następnie przystępujemy do konfiguracji *nginx*. Należy wejść do katalogu z aplikacją *nginx*, następnie w folder *conf* i otworzyć do edycji w dowolnym edytorze tekstowym plik *nginx.conf*. Wewnątrz znajduje się przykładowa konfiguracja aplikacji, większość linii stanowią komentarze. Nas interesuje dyrektywa "location /" dla serwera nasłuchującego na porcie 80:

```
location / {
    root    html;
    index  index.html index.htm;
}
```

Domyślna konfiguracja wskazuje na folder *html* w katalogu instalacyjnym *nginx*. To w nim będą wyszukiwane pliki do zwrócenia. Aby zmienić katalog dla głównej lokalizacji serwera należy zmodyfikować dyrektywę `root` podając jako jej wartość ścieżkę do utworzonego katalogu *webgis*:

```
location / {
    root    C:\Users\\Desktop\webgis;
    index  index.html index.htm;
}
```

Następnie należy uruchomić lub (jeśli jest już uruchomiony) zrestartować *nginx*. Po wejściu na stronę <http://127.0.0.1> (lub <http://localhost>) powinna pojawić się nowa strona.



Serwer pośredniczący

Częstym wykorzystaniem aplikacji typu *Apache* lub *nginx* są tzw. serwery pośredniczące (*proxy servers*). W takim wypadku serwer WWW bezpośrednio nie przetwarza żądań, a jedynie przekazuje je do innych aplikacji w celu wykonania operacji i zwrócenia wyniku. Jest to szczególnie przydatne gdy korzystamy z bazy danych i łączenie z nią oraz przetwarzanie danych odbywa się w dedykowanej aplikacji. Służy do tego dyrektywa `proxy_pass`.

```
location / {  
    proxy_pass http://localhost:8000;  
}
```

W powyższym przykładzie na serwerze, poza *nginx*, uruchomiona jest aplikacja obsługująca port 8000. Każde przychodzące żądanie (np. <http://strona.pl>) przekazywane jest przez *nginx* do tej aplikacji. Po zakończeniu odpowiedź wraca (za pośrednictwem *nginx*) do aplikacji klienckiej.

Ćwiczenie

Treść zadania

Utwórz konfigurację w *nginx* dla serwera obsługującego port 80, dzięki której możliwe będzie pośredniczenie *nginx* w komunikacji z *Geoserverem*. Adres do *Geoservera* powinien być określony jako `/geoserver`.

Opis

Należy wejść do katalogu z aplikacją *nginx*, następnie w folder *conf* i otworzyć do edycji w dowolnym edytorze tekstowym plik *nginx.conf*. Zgodnie z treścią dostęp do *Geoservera* ma odbywać się poprzez adres `/geoserver` w związku z tym w ustawieniach serwera dla portu 80 należy dodać nową dyrektywę `location`:

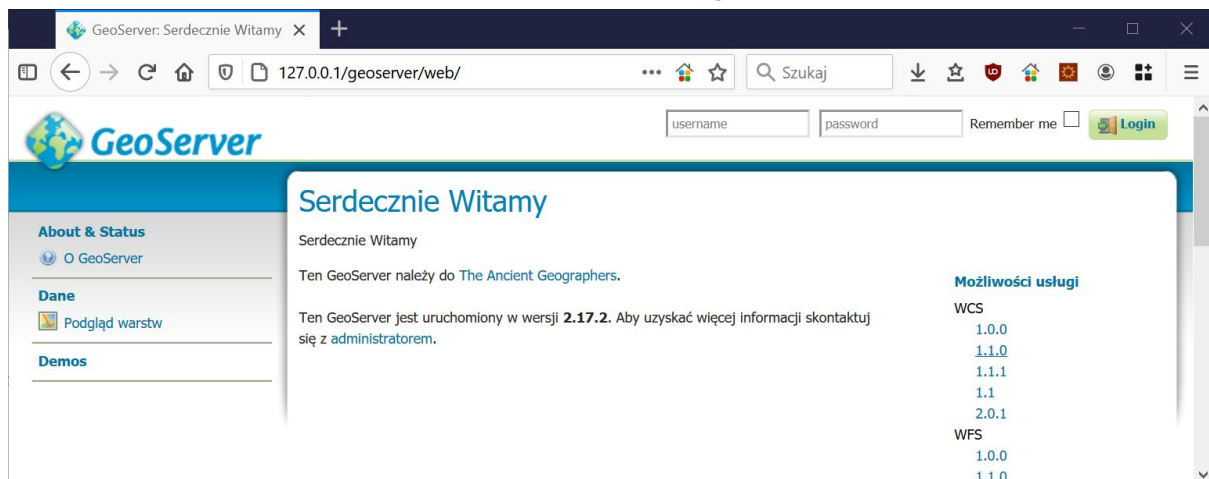
```
location /geoserver {  
}
```

Wewnątrz dyrektywy należy ustawić przekierowanie zapytań do *Geoservera* za pomocą dyrektywy `proxy_pass`. *Geoserver* domyślnie uruchomiony jest pod adresem

<http://127.0.0.1:8080>, a dostęp do panelu administracyjnego i usług odbywa się poprzez adres <http://127.0.0.1:8080/geoserver>. Do przekierowań należy podać drugi adres:

```
location /geoserver {
    proxy_pass http://127.0.0.1:8080/geoserver;
}
```

Następnie należy uruchomić lub (jeśli jest już uruchomiony) zrestartować *nginx*. Od tej pory wszystkie adresy <http://127.0.0.1/geoserver> będą przekierowywane do <http://127.0.0.1:8080/geoserver>, czyli do *Geoservera*. Można to zweryfikować wchodząc do panelu administratora pod adres <http://127.0.0.1:8080/geoserver/web>.



OpenLayers

OpenLayers to biblioteka JavaScript do wyświetlania danych przestrzennych na stronie internetowej. Posiada ona bogate możliwości związane z obsługą różnych formatów (rastrowych i wektorowych), stylizacji danych oraz interakcji z użytkownikiem. Jest ona dostępna na licencji Open Source i można ją dowolnie wykorzystywać, również w zastosowaniach komercyjnych. Posiada bogatą dokumentację oraz wiele przykładów pokazujących jak z niej korzystać.

Dane wyświetlane są domyślnie w odwzorowaniu *Web Mercator* (EPSG:3857). Jest to aktualnie standard wykorzystywany przez największe serwisy mapowe (*OpenStreetMap*, *Google Maps*, *Bing Maps*, *ESRI* itp.) reprezentujące dane dla całego świata. *OpenLayers* pozwala zmienić ten układ na inny, jednak wymaga to zainstalowania rozszerzenia *proj4js*. Na potrzeby szkolenia pozostawimy odwzorowanie domyślne.

Instalacja biblioteki

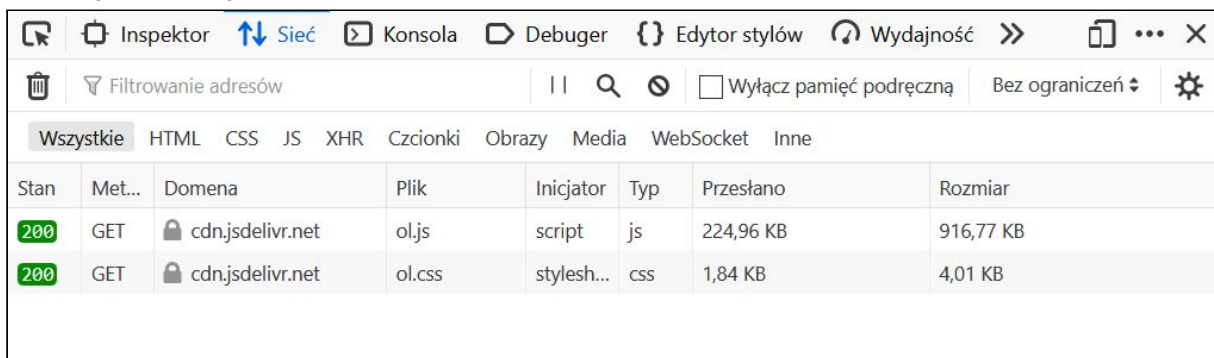
Biblioteka *OpenLayers* składa się z dwóch plików:

- **ol.js** - główny plik z kodem JavaScript,
- **ol.css** - dodatkowy plik zawierający style CSS dla wyświetlanych elementów.

Oba pliki można zapisać lokalnie w katalogu aplikacji jako pliki statyczne lub skorzystać z ich wersji udostępnianych poprzez internet. My skorzystamy z drugiego sposobu, w sekcji <HEAD> strony należy podać odnośniki do obu plików:

```
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/gh/openlayers/openlayers.github.io@master
/en/v6.4.3/css/ol.css" type="text/css">
<script
src="https://cdn.jsdelivr.net/gh/openlayers/openlayers.github.io@master/
en/v6.4.3/build/ol.js"></script>
```

Po odświeżeniu strony oba pliki powinny zostać pobrane przez przeglądarkę, co można zobaczyć w narzędziach deweloperskich:



The screenshot shows the Chrome DevTools Network tab. Two requests are visible: 'ol.js' (224,96 KB) and 'ol.css' (1,84 KB). Both requests have a status of 200 and were initiated by a script.

Stan	Met...	Domena	Plik	Inicjator	Typ	Przesłano	Rozmiar
200	GET	cdn.jsdelivr.net	ol.js	script	js	224,96 KB	916,77 KB
200	GET	cdn.jsdelivr.net	ol.css	styleh...	css	1,84 KB	4,01 KB

Po zainstalowaniu biblioteki *OpenLayers* w kodzie JavaScript dostępny będzie specjalny obiekt `ol`, z poziomu którego możliwy jest dostęp do API tej biblioteki.

Ćwiczenie

Treść zadania

W katalogu aplikacji internetowej utwórz nowy plik tekstowy o nazwie *script.js* i dodaj do niego referencję w pliku *index.html*.

Opis

W katalogu roboczym należy utworzyć pusty plik o nazwie *script.js*. Następnie w pliku *index.html* należy go osadzić aby kod w nim zawarty mógł być wykonywany na stronie internetowej. W tym celu należy wykorzystać tag `<SCRIPT>`, nazwę pliku JS podaje się jak atrybut `src`. W związku z tym, że plik jest pobierany z tego samego adresu co plik *index.html* można pominąć domenę i podać tylko nazwę samego pliku.

Plik należy dodać na końcu sekcji `<BODY>` ponieważ powinien on być wykonany po wczytaniu struktury strony.

```
<body>
  <h2>Strona główna</h2>
  <script src="script.js"></script>
</body>
```

Dodanie mapy na stronie internetowej

Aby umieścić kontrolkę z mapą na stronie należy stworzyć na stronie element `<DIV>`, który jest umieszczony w wybranym miejscu. Aby *OpenLayers* wiedział, do którego elementu ma się podłączyć należy określić unikalny identyfikator (atrybut `id`):

```
<body>
  <h2>Strona główna</h2>
  <div id="mapa"></div>
  <script src="script.js"></script>
</body>
```

Następnie należy w pliku *script.js* stworzyć specjalny obiekt `ol.Map`, który będzie reprezentował mapę:

```
var map = new ol.Map({
  target: 'mapa'
});
```

Parametr `target` określa element `<DIV>`, w który zostanie wstawiona mapa.

Widok mapy

Aby poprawnie wyświetlić dane po wczytaniu strony należy określić inicjalny widok mapy czyli punkt środkowy i poziom przybliżenia mapy. Służy do tego obiekt `ol.View`, który przyjmuje obie informacje jako parametry, odpowiednio `center` i `zoom`. Sam widok jest definiowany wewnątrz obiektu mapy jako parametr `view`.

```
var map = new ol.Map({
  target: 'mapa',
  view: new ol.View({
    center: ol.proj.fromLonLat([20, 52.2]),
    zoom: 6
  })
});
```

Funkcja `fromLonLat` pozwala przeliczyć współrzędne podawane w układzie *WGS 84* na inny układ. Domyślnym układem docelowym jest *Web Mercator* (EPSG:3857). Poziom zbliżenia podaje się jako liczbę naturalną, im wyższa wartość tym większe przybliżenie.

Dodawanie warstw

OpenLayers wspiera wiele typów warstw, zarówno rastrowych jak i wektorowych. Dane przestrzenne muszą być udostępnione w sieci w formie usługi (*WMS*, *WMTS*, *WFS*) lub pliku do pobrania (*GeoJSON*, *GML 2/3*).

Definiując warstwę w *OL* należy podać dwie główne klasy. Pierwsza z nich znajduje się w module `ol.layer` i definiuje rodzaj warstwy i pozwala bibliotece prawidłowo wyświetlić elementy na mapie (m.in. styl dla danych wektorowych). Druga to obiekt z modułu `ol.source` i w tym miejscu podawane są informacje o źródle danych czyli np. adres URL. Aby wyświetlić warstwy należy w obiekcie mapy określić parametr `layers`. Jest to tablica JavaScript, w której będziemy podawać kolejne warstwy:

```
var map = new ol.Map({
  ...,
  layers: [ ],
});
```

Rastrowe warstwy kafłowe

Jako pierwszą dodamy podkładową warstwę rastrową. Warstwy podkładowe służą głównie do podglądu i lokalizacji innych elementów przestrzennych. Dzięki nim użytkownik może określić jaki obszar aktualnie widzi i ewentualnie przenieść się do innego miejsca. Są zazwyczaj zlokalizowane na samym dole mapy i inne warstwy mogą je zasłaniać. Najczęściej wykorzystywane są dane udostępniane przez zewnętrzne serwisy np. *OpenStreetMap*, *Google Maps* czy *Geoportal.gov.pl*. *OpenLayers* posiada specjalną klasę ułatwiającą definiowanie źródła dla *OpenStreetMap* więc jako pierwszą warstwę dodamy mapę z tego projektu. Będzie to kafłowa warstwa rastrowa.

Rastrowe mapy w formie kafli definiowane są za pomocą klasy `ol.layer.Tile`, jako źródło mapy należy podać klasę `ol.source.OSM`.

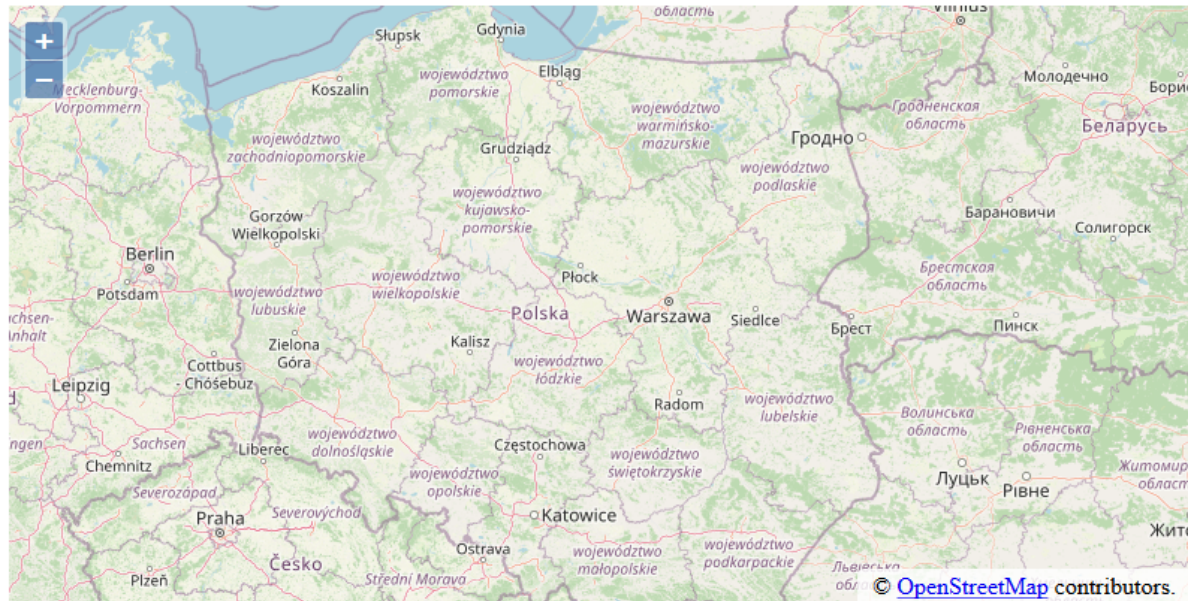
```
var osm = new ol.layer.Tile({
  source: new ol.source.OSM()
});
```

Zmienna `osm` przechowuje teraz referencję do warstwy. Aby ją wyświetlić należy podać tą zmienną we wcześniej przygotowanej liście:

```
var map = new ol.Map({
  ...,
  layers: [ osm ],
});
```

W ten sposób dodaliśmy pierwszą warstwę do mapy i efekt można zobaczyć w przeglądarce:

Strona główna



W podglądzie pobieranych plików można zauważyć, że aby wyświetlić warstwę *OpenLayers* pobrał kilka plików graficznych. Każdy z nich określa prostokątny fragment przestrzeni tzw. kafelki (*tile*). W trakcie wczytywania danych pojawiają się one na mapie stopniowo, widać wtedy dokładnie jak jest tworzona warstwa. Po wyświetleniu wszystkich elementów na mapie, użytkownik widzi je jako pojedynczą warstwę.

200	GET	localhost	/	Browse...	html	839 B	601 B
200	GET	localhost	script.js	script	js	710 B	459 B
200	GET	cdn.jsdelivr.net	ol.css	stylesh...	css	1,81 KB	4,01 KB
200	GET	cdn.jsdelivr.net	ol.js	script	js	224,98 KB	916,77 KB
200	GET	c.tile.openstreetmap...	21.png	ol.js:8 (...	png	52,13 KB	51,49 KB
200	GET	b.tile.openstreetma...	20.png	ol.js:8 (...	png	40,17 KB	39,54 KB
200	GET	a.tile.openstreetma...	21	https://b.tile.openstreetmap.org/6/35/20.png			43,23 KB
200	GET	c.tile.openstreetmap...					48,14 KB
200	GET	b.tile.openstreetma...					53,70 KB
200	GET	a.tile.openstreetma...					35,94 KB
404	GET	localhost					153 B

256 × 256

WMS

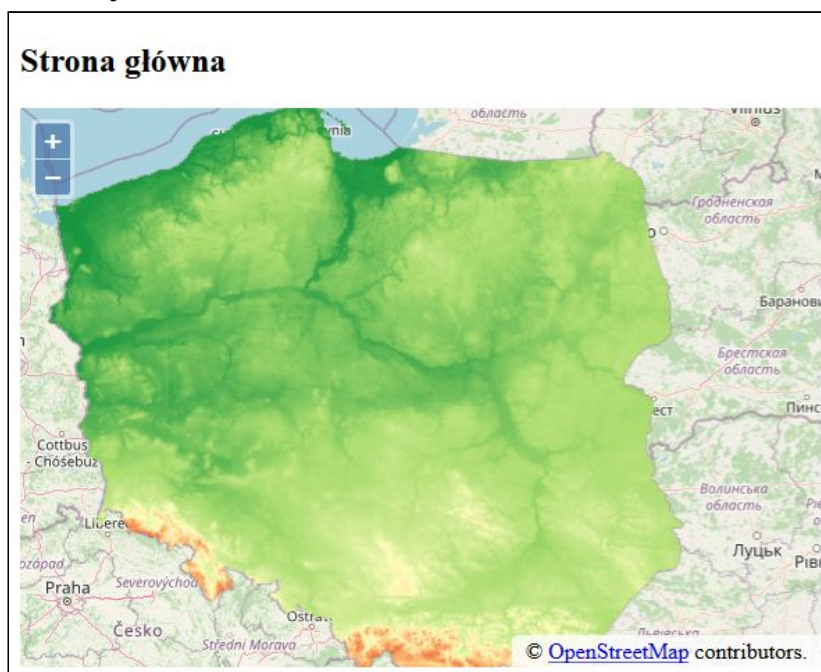
Warstwy WMS pobierane są jako pojedynczy obraz reprezentujący dany obszar. Aby wyświetlić ten typ warstwy w *OpenLayers* należy skorzystać z klasy `ol.layer.Image`. Jako źródło podaje się natomiast klasę `ol.source.ImageWMS`. Przyjmuje ona kilka parametrów, najważniejsze z nich to `url` (adres do usługi WMS) oraz `params`. Ten ostatni jest obiektem JavaScript, w którym można określić parametry zgodne ze specyfikacją WMS dla operacji *GetMap*, najczęściej podaje się nazwy warstw (`layers`), które mają być wyświetlone, ewentualnie styl (`style`) jeśli jest inny niż domyślny. Pozostałe parametry ustawia *OpenLayers*.

```
var nmt = new ol.layer.Image({
  source: new ol.source.ImageWMS({
    url: 'http://127.0.0.1:8080/geoserver/webgis/ows',
    params: {
      layers: 'nmt'
    }
  })
});
```

Mając zmienną `nmt` można ją przekazać do tablicy z listą warstw do wyświetlenia na mapie:

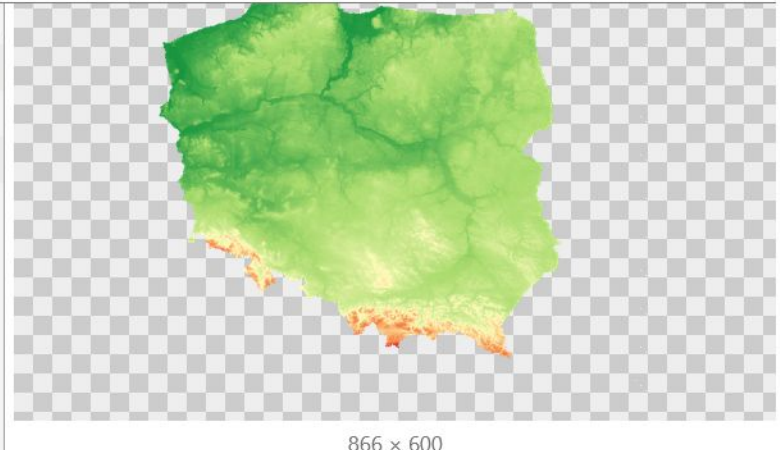
```
var map = new ol.Map({
  ...,
  layers: [ osm, nmt ],
});
```

Kolejność elementów w tablicy `layers` ma znaczenie. Pierwsza warstwa jest wyświetlana na samym dole, druga wyżej, trzecia jeszcze wyżej itd. Dzięki temu sterujemy kolejnością wyświetlania warstw na mapie. Po odświeżeniu strony powinna pojawić się nowa warstwa, nad mapą podkładową OSM:



Warto zwrócić uwagę na pobierane dane. Dla warstwy WMS jest wykonywane tylko jedno żądanie, a parametr *LAYERS* w adresie URL został ustawiony przez nas w kodzie. Pozostałe parametry wypełnia *OpenLayers*:

Stan	Met...	Domena	Plik	Inicjator	Typ	Przesłano	Rozmiar	
200	GET	localhost	/	Browse...	html	839 B	601 B	
200	GET	localhost	script.js	script	js	966 B	715 B	
200	GET	cdn.jsdelivr.net	ol.css	stylesh...	css	1,84 KB	4,01 KB	
200	GET	cdn.jsdelivr.net	ol.js	script	js	224,96 KB	916,77 KB	
200	GET	127.0.0.1:8080	ows?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetMap&LAYERS=dem84	ol.js:8 (...)	png	94,23 KB	94,04 KB	
200	GET	c.tile.o	http://127.0.0.1:8080/geoserver/test/ows?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetMap&FORMAT=image%2Fpng&TRANSPARENT=true&LAYERS=dem84&CRS=EPSG%3A3857&STY...					
200	GET	b.tile.o						
200	GET	a.tile.o						
200	GET	c.tile.openstreetmap...						
200	GET	b.tile.openstreetma...						
200	GET	a.tile.openstreetma...						
404	GET	localhost						



Ćwiczenie

Treść zadania

Dodaj do mapy warstwy *kondracki*, *województwa* i *miasta* z *Geoserver* oraz warstwę *Ortofotomapa standardowa* ze strony *geoportal.gov.pl*. Spis usług przeglądania dla *geoportal.gov.pl* można znaleźć na stronie <https://www.geoportal.gov.pl/uslugi/usluga-przeglądania-wms>.

Opis

Dodawanie warstw wektorowych i zwracanie ich jako rastry z usługi *WMS* nie różni się od dodania warstwy rastrowej z numerycznym modelem terenu. Należy skorzystać z klas *ol.layer.Image* i *ol.source.ImageWMS*, jako adres podać adres usługi *WMS* (<http://127.0.0.1:8080/geoserver/webgis/ows>), podając jako parametr *layers* odpowiednie nazwy (zgodnie z tym jak są zdefiniowane w *Geoserver*).

```
var kondracki_wms = new ol.layer.Image({
  source: new ol.source.ImageWMS({
    url: 'http://127.0.0.1:8080/geoserver/webgis/ows',
    params: { layers: 'kondracki' }
  })
});
```



```


    })
  });

  var wojewodztwa_wms = new ol.layer.Image({
    source: new ol.source.ImageWMS({
      url: 'http://127.0.0.1:8080/geoserver/webgis/ows',
      params: { layers: 'wojewodztwa' }
    })
  });

  var miasta_wms = new ol.layer.Image({
    source: new ol.source.ImageWMS({
      url: 'http://127.0.0.1:8080/geoserver/webgis/ows',
      params: { layers: 'miasta' }
    })
  });

  var map = new ol.Map({
    target: ...,
    view: ...,
    layers: [ osm, nmt, kondracki_wms, wojewodztwa_wms, miasta_wms ]
  });

```

Aby dodać warstwę z ortofotomapą udostępnioną przez *geoportal.gov.pl* należy wejść na stronę <https://www.geoportal.gov.pl/uslugi/usluga-przegladania-wms> i znaleźć odpowiednią warstwę. Następnie skopiować adres usługi. Będzie on użyty w klasie `ol.source.ImageWMS` jako parametr `url`. Potrzebna jest jednak jeszcze nazwa warstwy, która zostanie użyta w parametrze `layers`. Można do tego wykorzystać *QGIS*. W tym celu należy otworzyć okno *Zarządzanie źródłami danych* wybierając menu *Warstwa -> Zarządzanie źródłami danych* lub przez odpowiednią ikonę na pasku narzędzi . Następnie przejść do zakładki *WMS/WMTS* i dodać nowe połączenie klikając przycisk *Nowa*. W nowym oknie podajemy nazwę, adres usługi WMS i klikamy *OK*.

Utwórz nowe połączenie WMS/WMTS

Szczegóły połączenia

Nazwa: Ortofotomapa
 URL: s/service/img/guest/ORTO/MapServer/WMSServer

Uwierzytelnianie

Konfiguracja: Bez zabezpieczeń

Wybierz lub utwórz konfigurację uwierzytelniania

Bez uwierzytelniania

Konfiguracja przechowuje zaszyfrowane dane w bazie danych uwierzytelniania QGIS.

Opcje WMS/WMTS

Referer:

Tryb DPI: wszystko

Ignoruj GetMap/GetTile URI podany w Capabilities
 Ignoruj GetFeatureInfo URI podany w Capabilities
 Ignoruj orientację osi (WMS 1.3/WMTS)
 Ignoruj zgłoszone zakresy warstw
 Odwróć orientację osi
 Wygładź przekształcenie bitmap

OK Anuluj Pomoc

Nowe połączenie powinno być dostępne na liście, po kliknięciu *Połącz* pojawi się lista dostępnych warstw, z której można odczytać nazwę warstwy zawierającej ortofotomapę czyli *Raster*.

Warstwy Kolejność warstw Kafle Szukaj serwera

Ortofotomapa

Połącz Nowa Edytuj Usuń Wczytaj Zapisz Dodaj domyślne serwery

ID	Nazwa	Tytuł	Streszczenie
0		Ortofotomapy ...	
1		ORTOFOTOMAPA	ORTOFOTOMAPA
2	Raster	Raster	

Teraz można dodać nową warstwę *WMS* do naszej strony, analogicznie jak inne warstwy.

```
var ortofotomapa = new ol.layer.Image({
  source: new ol.source.ImageWMS({
    url: 'https://mapy.geoportal.gov.pl/wss/service/img/guest/ORTO/MapServer/WMServer',
    params: { layers: 'Raster' }
  })
})
```

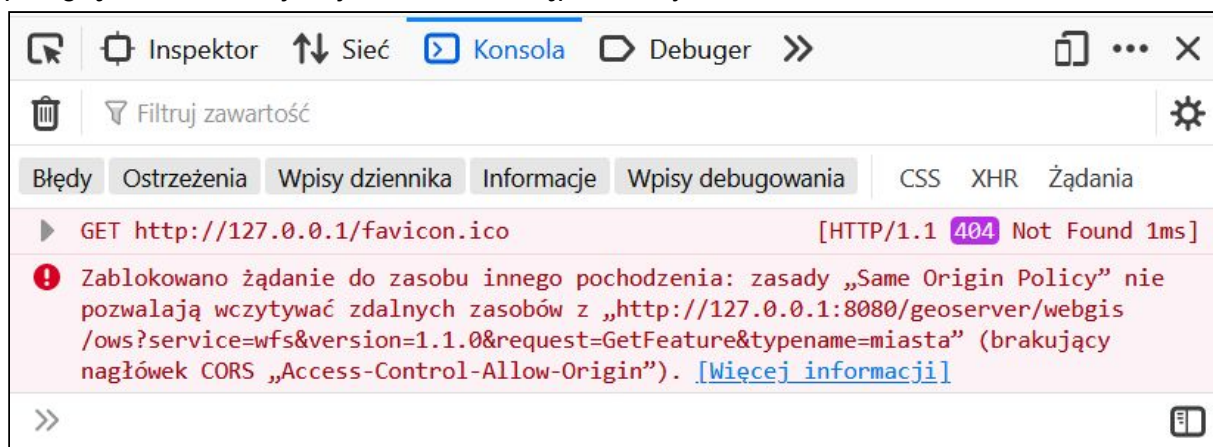
```
});

var map = new ol.Map({
  target: ...,
  view: ...,
  layers: [ osm, nmt, kondracki_wms, ortofotomapa, wojewodztwa_wms,
miasta_wms ]
});
```

Dane wektorowe

Wyświetlanie danych wektorowych działa inaczej niż rastrowych, gdzie pobierany i wyświetlany na mapie jest gotowy obraz. W takim wypadku dane w formie wektorowej najpierw muszą zostać pobrane, a następnie wyrenderowane jako obraz, z użyciem wskazanego stylu. Jednak ze względu na mechanizmy bezpieczeństwa jakie stosuje przeglądarka może to być utrudnione. Wynika to z reguły tego samego pochodzenia (*same-origin policy*) określającej sposób w jaki skrypty jednego pochodzenia (*origin*) mogą komunikować się z zasobami innego pochodzenia. Elementy są tego samego pochodzenia jeśli protokół, domena i port w adresie URL są jednakowe. Przykładowo dla strony <https://gis-support.pl> zasób z adresu <https://gis-support.pl/szkolenia-gis/> jest tego samego pochodzenia, ale już adres <http://gis-support.pl> lub <https://gis-support.pl:81> nie.

Są pewne wyjątki, które pozwalają na osadzania zasobów międzyźródłowych na stronie. Przykładem jest dostęp do pobranych obrazów, dzięki czemu wyświetlanie danych rastrowych (*WMS*, *WMTS*) nie sprawia problemów w *OpenLayers*. Innymi wyjątkami są m.in. pliki *JavaScript* (osadzone w znaczniku `<SCRIPT>`) i *CSS* (`<LINK>`) oraz *media* (`<VIDEO>`). Biorąc powyższe pod uwagę, jeśli w *OpenLayers* chcemy wyświetlić dane pochodzące z innego serwera (lub tego samego serwera, ale dostępnego pod inną domeną) to przeglądarka zablokuje tej bibliotece dostęp do danych.



Ograniczenia wynikające z reguły tego samego pochodzenia można obejść na kilka sposobów:

- wykorzystanie serwera WWW jako proxy do zewnętrznych zasobów, dzięki czemu będą one dostępne pod tym samym adresem,

- wykorzystanie międzyźródłowego dostępu do zasobów (*cross-origin resource sharing, CORS*), wymaga odpowiedniej konfiguracji serwisu, z którego pobierane są dane.

W naszym przypadku skorzystamy z pierwszej metody, ponieważ *nginx* został już skonfigurowany, tak aby *Geoserver* działał na tej samej domenie co tworzona strona internetowa. Z tego powodu do danych wektorowych wykorzystamy adres <http://127.0.0.1:80>.

Druga metoda wymaga zmodyfikowania plików konfiguracyjnych *Geoserver* i jest opisana w dokumentacji

<https://docs.geoserver.org/latest/en/user/production/container.html#enable-cors>. Jednak ze względu na to, że nie zawsze mamy możliwość konfigurowania zewnętrznych serwisów, pierwsza metoda jest bardziej uniwersalna.

Aby dodać dane wektorowe w *OpenLayers* należy skorzystać z klasy `ol.layer.Vector`, a do definiowania źródeł z `ol.source.Vector`.

```
var warstwa = new ol.layer.Vector({
  source: new ol.source.Vector({
    ...
  }),
});
```

Wewnątrz klasy źródła należy określić obiekty przestrzenne, które będą wyświetlone na mapie. Można to zrobić na kilka sposobów, m.in.:

- podając adres (atrybut `url`) i format danych (`format`),
- podając kolekcję obiektów, czyli instancji klasy `ol.Feature` (`features`),
- podając specjalną funkcję, w której odbywa się pobranie i parsowanie danych (`loader`).

Skupimy się na pierwszej, najprostszej metodzie. Pozwala ona pobrać dane bezpośrednio z *Geoserver* i je wyświetlić. Aby uzyskać adres danych w konkretnym formacie należy w panelu administracyjnym otworzyć *Podgląd warstw*, wyszukać warstwę i z listy Wszystkie formaty wybrać format. Zostanie otworzone nowe okno przeglądarki z adresem, dzięki któremu możliwe jest pobranie zasobu w określonym formacie. Można go skopiować i wykorzystać w parametrze `url`. Warto przyjrzeć się całemu adresowi ponieważ możliwe, że będzie on wymagał korekty np. jeśli chcemy pobrać całą warstwę, a w URL jest parametr `maxFeatures=50`, to zostanie pobranych tylko 50 pierwszych obiektów. W takim wypadku można usunąć go z adresu. Parametry w URL są związane ze specyfikacją danej usługi, wybrane parametry z jakimi się można spotkać to m.in.:

- `service` - określa rodzaj usługi, dla pobierania danych wektorowych jest to *WFS*,
- `version` - wersja usługi, dla *WFS Geoserver* wspiera 1.0.0 (domyślnie zwracany jest *GML 2*), 1.1.0 (*GML 3.1*) oraz 2.0 (*GML 3.2*),
- `request` - rodzaj operacji, pobranie odbywa się za pomocą `GetFeature`,
- `typeName` (*WFS 1.0 i 1.1*) i `typeNames` (*WFS 2.0*) - nazwy typów obiektów, które zostaną pobrane, w przypadku danych z *Geoservera* podaje je w formie: `obszar_roboczy:warstwa`,

- `outputFormat` - zwracany format danych np. GML2, GML3, GML32, `application/json` (dla *GeoJSON*),
- `srsName` - określa układ współrzędnych w jakim zostaną zwrócone geometrie, transformacja odbywa się po stronie serwera, bez podania tego parametru dane zwracane są w układzie źródła danych, dla *WFS 1.0.0* przyjmuje on format `http://www.opengis.net/gml/srs/epsg.xml#XXXX`, dla *WFS 1.1.0* i *2.0* `urn:x-ogc:def:crs:EPSG:XXXX`, a w przypadku *GeoJSON* `EPSG:XXXX`, gdzie `XXXX` oznacza kod EPSG układu.

Częstym problemem jest brak wyświetlanych danych na mapie pomimo, że zostały one pobrane z serwera, a jednym z powodów jest złe określenie układu współrzędnych. *OpenLayers* automatycznie obsługuje układ *WGS 84* (EPSG:4326) oraz *Web Mercator* (EPSG:3857) i potrafi dokonywać automatycznej transformacji pomiędzy nimi. Do obsługi innych układów wymagane jest zainstalowanie i skonfigurowanie biblioteki *proj4js*.

Jako format należy podać klasę z modułu `ol.format`, dostępne są m.in:

- `ol.format.WFS` - pobieranie danych z usługi *WFS*, aktualnie (*OpenLayers 6.4.3*) wspierane są wersje 1.0.0 i 1.1.0, ale w najnowszej wersji biblioteki *OpenLayers* zostanie dodane wsparcie dla *WFS 2.0* (źródło: <https://github.com/openlayers/openlayers/issues/11410>).

```
var miasta_wfs = new ol.layer.Vector({
  source: new ol.source.Vector({
    format: new ol.format.WFS({
      srsName: "EPSG:4326",
      featureType: "miasta",
      featureNS: "http://localhost/webgis",
    }),
    url:
'/geoserver/webgis/ows?service=WFS&version=1.1.0&request=GetFeature&type
Name=webgis%3Amiasta&outputFormat=gml3'
  }),
});
```

- `ol.format.GML2`, `ol.format.GML3` i `ol.format.GML32` - pobieranie danych w formacie *GML*, odpowiednio w wersjach 2, 3.1 i 3.2.

```
var wojewodztwa_gml = new ol.layer.Vector({
  source: new ol.source.Vector({
    format: new ol.format.GML({
      srsName: "EPSG:3857",
      featureType: "wojewodztwa",
      featureNS: "http://localhost/webgis"
    }),
    url:
```

```
'/geoserver/webgis/ows?service=WFS&version=1.1.0&request=GetFeature&typeName=webgis%3Awojewództwa&outputFormat=GML3&srsName=urn:x-ogc:def:crs:EPSG:3857'  
  }),  
});
```

- `ol.format.GeoJSON` - obsługa formatu *GeoJSON*. Wg oficjalnej specyfikacji geometria

```
var kondracki_geojson = new ol.layer.Vector({  
  source: new ol.source.Vector({  
    format: new ol.format.GeoJSON(),  
    url:  
    '/geoserver/webgis/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=webgis%3Akondracki&outputFormat=application%2Fjson&srsName=EPSG:4326'  
  }),  
});
```

Po zdefiniowaniu warstw należy je dodać do listy warstw mapy *OL*:

```
var map = new ol.Map({  
  target: ...,  
  view: ...,  
  layers: [ ..., wojewodztwa_gml, kondracki_geojson, miasta_wfs ]  
});
```

Rozszerzenia OpenLayers

Możliwe jest dodawanie nowych funkcjonalności *OpenLayers* dzięki systemowi rozszerzeń. Są to biblioteki JavaScript, które należy zdefiniować w nagłówku strony. Pozwalają np. na obsługę wielu układów współrzędnych, łatwe dodawanie przycisków do mapy czy tworzenie dynamicznej legendy.

Jednym z ciekawszych rozszerzeń jest *ol-ext*, które pozwala dodać wiele elementów do mapy. Na stronie projektu (<https://viglino.github.io/ol-ext/>) jest wiele przykładów obrazujących jego możliwości. Aby dodać *ol-ext* do strony należy w sekcji <HEAD> strony dodać definicje pliku JavaScript i CSS. Trzeba to zrobić za definicją biblioteki *OL*, ponieważ jest ona wymagana przez to rozszerzenie:

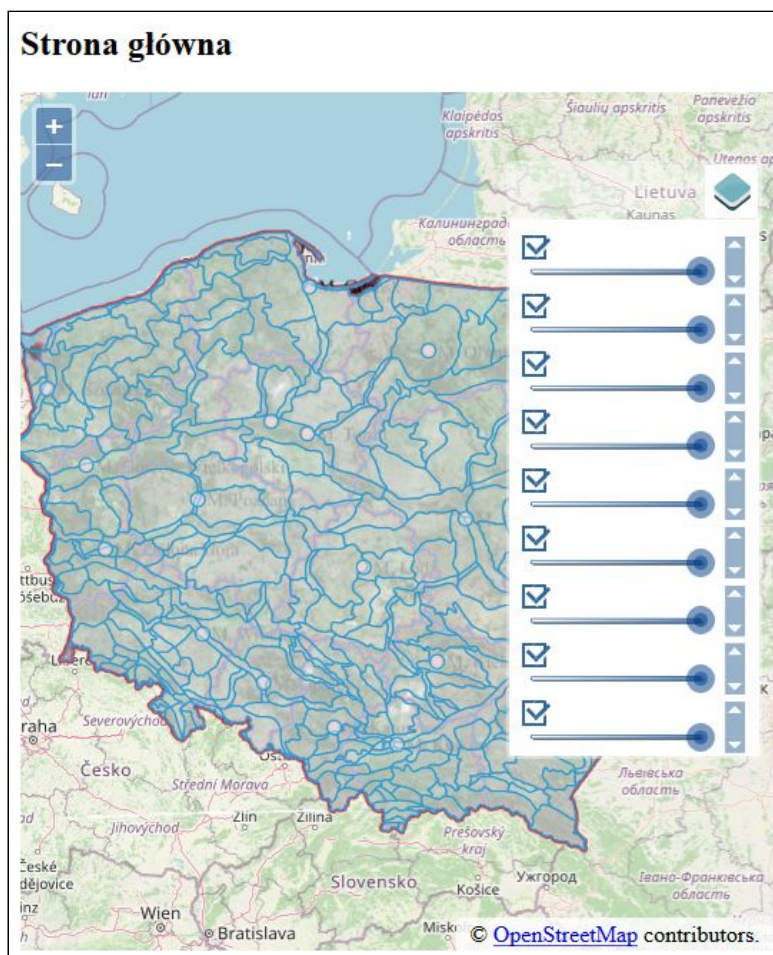
```
<link rel="stylesheet"  
href="https://cdn.rawgit.com/Viglino/ol-ext/master/dist/ol-ext.min.css"  
>  
<script type="text/javascript"  
src="https://cdn.rawgit.com/Viglino/ol-ext/master/dist/ol-ext.min.js"></script>
```


Legenda mapy

Korzystając z *ol-ext* bardzo łatwo można dodać nowe elementy do mapy. Przykładem jest legenda pozwalająca określać widoczność warstw oraz ich kolejność. Służy do tego klasa `ol.control.LayerSwitcher()`. Pozwala ona stworzyć nowy kontroler. W nomenklaturze *OpenLayers* kontroler (*control*) jest to dowolna kontrolka, którą widzi użytkownik np. przycisk, legenda lub okno z atrybutami. Aby dodać kontroler do mapy należy wykorzystać metodę `addControl` z obiektu mapy.

```
var legenda = new ol.control.LayerSwitcher();
map.addControl( legenda );
```

Legenda będzie dostępna w górnym prawym rogu w formie przycisku, po którego kliknięciu rozwinię się lista warstw:

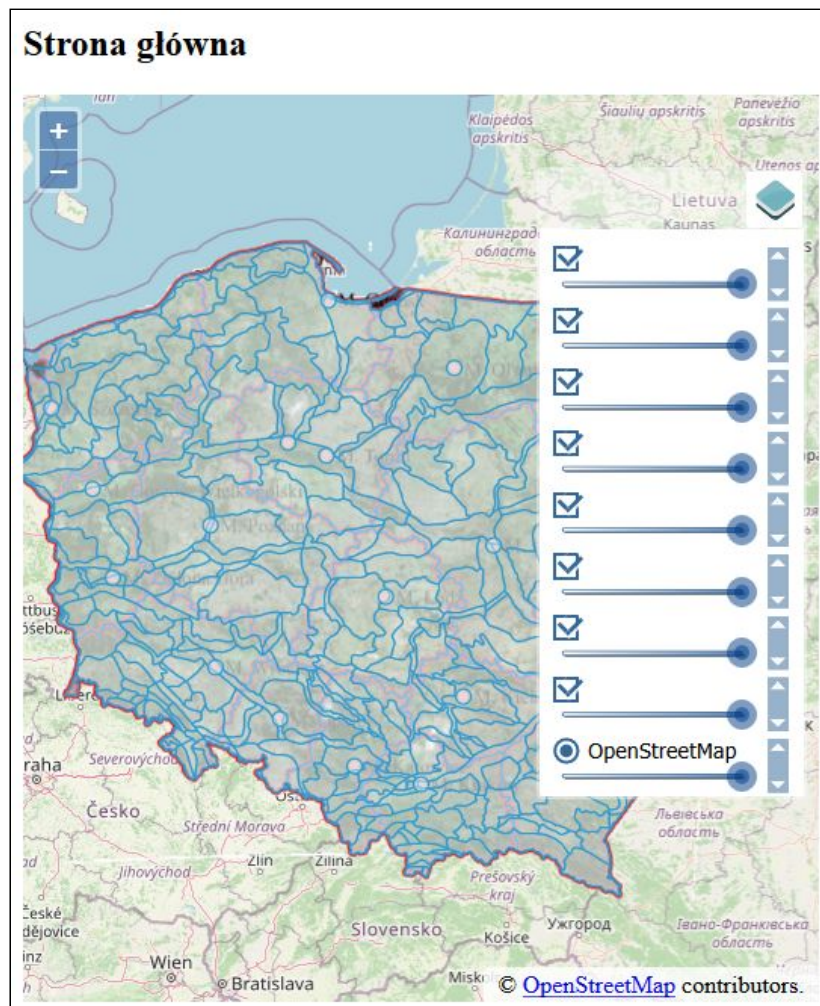


Jak widać brakuje na niej opisów poszczególnych pozycji. Aby je dodać należy dodać dodatkowe informacje w klasach warstw. Częścią opcji dla danej warstwy można sterować poprzez dodatkowe parametry dodawane przy jej tworzeniu, z których korzysta *ol-ext*:

- **title** lub **name** - nazwa warstwy wyświetlana w legendzie,
- **displayInLayerSwitcher** - wartość logiczna określająca czy warstwa ma być wyświetlona w legendzie, należy ustawić na `false` jeśli nie,

- **baseLayer** - wartość logiczna określająca czy dana warstwa jest warstwą bazową, tylko jedna warstwa bazowa może być widoczna w danym momencie,
- **visible** - wartość logiczna oznaczająca czy warstwa ma być widoczna po załadowaniu strony,
- **displayInLayerSwitcher** - wartość logiczna, jeśli ustawiona na `true` to warstwa nie będzie widoczna w legendzie.

```
var osm = new ol.layer.Tile({
  source: ...,
  title: 'OpenStreetMap',
  baseLayer: true
});
```



Parametr `title` został wyświetlony jako nazwa w legendzie, natomiast ustawienie `baseLayer` na `true` zmieniło kontrolkę z pola wyboru na pole opcji. Jeśli kilka warstw jest oznaczona jako bazowe to możliwe będzie wskazanie tylko jednej z nich.

Ćwiczenie

Treść zadania

Dodaj opisy do zdefiniowanych warstw zgodnie z poniższą listą:

- **ortofotomapa** - nazwa *Ortofotomapa*, warstwa bazowa, domyślnie wyłączona,
- **nmt** - *Numeryczny model terenu*, warstwa bazowa, domyślnie wyłączona,
- **kondracki_wms** - *Podział fizjograficzny Polski WMS*
- **województwa_wms** - *Województwa WMS*, wyłączona
- **miasta_wms** - *Miasta wojewódzkie WMS*, zawsze na górze
- **województwa_gml** - *Województwa WFS*
- **kondracki_geojson** - *Podział fizjograficzny Polski GeoJSON*, wyłączona
- **miasta_wfs** - *Miasta wojewódzkie GML*, wyłączona

Kod źródłowy

```
var ortofotomapa = new ol.layer.Image({
  ...,
  title: 'Ortofotomapa',
  visible: false,
  baseLayer: true
});

var nmt = new ol.layer.Image({
  ...,
  title: 'Numeryczny model terenu',
  visible: false,
  baseLayer: true
});

var kondracki_wms = new ol.layer.Image({
  ...,
  title: 'Podział fizjograficzny Polski WMS'
});

var wojewodztwa_wms = new ol.layer.Image({
  ...,
  title: 'Województwa WMS',
  visible: false
});

var miasta_wms = new ol.layer.Image({
  ...,
  title: 'Miasta wojewódzkie WMS'
});

var miasta_wfs = new ol.layer.Vector({
  ...,
```

```

    title: 'Miasta wojewódzkie WFS',
    visible: false
  });

var wojewodztwa_gml = new ol.layer.Vector({
  ...,
  title: 'Województwa GML'
});

var kondracki_geojson = new ol.layer.Vector({
  ...,
  title: 'Podział fizjograficzny Polski GeoJSON',
  visible: false
});

```

