Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Unia Europejska Europejski Fundusz Społeczny

Protokoły TCP, UDP i HTTP

Technologie Internetowe

Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Protokół komunikacyjny

Protokół komunikacyjny to zbiór ścisłych reguł i kroków postępowania, które są automatycznie wykonywane przez urządzenia komunikacyjne w celu nawiązania łączności i wymiany danych

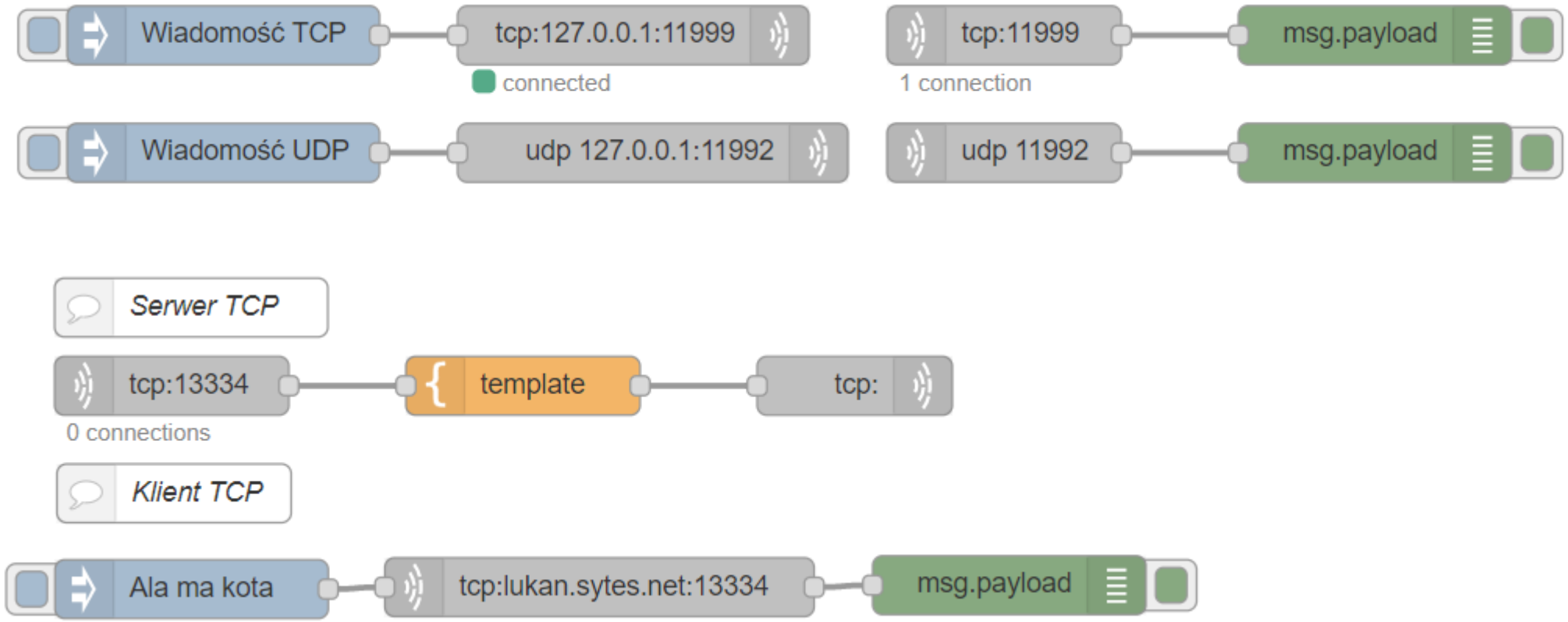
Projekt „SezAM wiedzy, kompetencji i umiejętności” jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Protokół TCP i UDP

- TCP (od ang. Transmission Control Protocol) – połączeniowy, niezawodny, strumieniowy protokół komunikacyjny stosowany do przesyłania danych między procesami uruchomionymi na różnych maszynach. Protokół TCP operuje w warstwie transportowej modelu OSI
- UDP (ang. User Datagram Protocol – protokół pakietów użytkownika) UDP stosowany jest w warstwie transportowej modelu OSI. Nie gwarantuje dostarczenia datagramu.

Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Realizacja połączenia TCP i UDP – w Node-RED



- http://lukan.sytes.net:1880/#flow/a3ae9ff5.5f423

Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Lokalne wysłanie i odbiór wiadomości TCP

```
Socket serverSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
Socket clientSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
Socket internalSocket;
byte[] recBuffer = new byte[256];
serverSocket.Bind(new IPEndPoint(IPAddress.Parse("127.0.0.1"), 1024));
serverSocket.Listen(1);
clientSocket.Connect("127.0.0.1", 1024);
internalSocket = serverSocket.Accept();
clientSocket.Send(ASCIIEncoding.ASCII.GetBytes("Hello world!"));
internalSocket.Receive(recBuffer);
Console.WriteLine(ASCIIEncoding.ASCII.GetString(recBuffer));
Console.Read();
```

Uruchamiać jako aplikację konsolową

Projekt „SezAM wiedzy, kompetencji i umiejętności” jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Wysłanie wiadomości UDP i TCP

```
System.Net.Sockets.UdpClient udpClient = new UdpClient();        // Klient UDP
byte[] dane = ASCIIEncoding.ASCII.GetBytes("Wiadomosc UDP");
udpClient.Send(dane, dane.Length, "lukan.sytes.net", 11992);
udpClient.Close();                                                // Koniec klienta UDP
TcpClient tcpClient = new TcpClient();                            // Klient TCP
tcpClient.Connect("lukan.sytes.net", 11999);                      // Próba połączenia
BinaryWriter writer = new BinaryWriter(tcpClient.GetStream());    // Dane do wysłania
writer.Write("Wiadomość TCP");
writer.Close();
tcpClient.Close();                                                // koniec klienta TCP
```

Odbiornikiem jest graf Node-RED przedstawiony 2 slajdy wcześniej

Projekt „SezAM wiedzy, kompetencji i umiejętności” jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Protokół HTTP

- HTTP (ang. *Hypertext Transfer Protocol*) protokół przesyłania dokumentów hipertekstowych.
- Protokół określa format żądania klienta wysyłanego na serwer i format odpowiedzi serwera.
- Żądanie poza adresem zwrotnym może zawierać parametry przekazywane do aplikacji serwerowej np. dane formularza przesłane metodą GET lub POST.
- Żądanie metodą GET można wysłać z dowolnej przeglądarki.
- Protokół jest bezstanowy – nie zapamiętuje stanu sesji z klientem.
- Specyfikacja protokołu w dokumencie RFC 2616.

Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Protokół HTTP Metody

Żądanie HTTP rozpoczyna się od podania nazwy metody. Metoda jest umowną nazwą działania, które powinno być wykonane po stronie serwera. Podstawowe metody HTTP to:

- GET – pobranie zasobu wskazanego przez URI, może mieć postać warunkową jeśli w nagłówku występują pola warunkowe takie jak "If-Modified-Since"
- HEAD – pobiera informacje o zasobie, stosowane do sprawdzania dostępności zasobu
- PUT – przyjęcie danych przesyłanych od klienta do serwera, najczęściej aby zaktualizować wartość encji
- POST – przyjęcie danych przesyłanych od klienta do serwera (np. wysyłanie zawartości formularzy)
- DELETE – żądanie usunięcia zasobu, włączone dla uprawnionych użytkowników
- OPTIONS – informacje o opcjach i wymaganiach istniejących w kanale komunikacyjnym
- TRACE – diagnostyka, analiza kanału komunikacyjnego
- CONNECT – żądanie przeznaczone dla serwerów pośredniczących pełniących funkcje tunelowania
- PATCH – aktualizacja części danych

Metody powinny być implementowane zgodnie z powyższym opisem, chociaż protokół tego nie wymusza. Stosowanie się do powyższych zaleceń jest dobrą praktyką programistyczną.

Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Żądanie HTTP

Po nazwie metody podawany jest identyfikator zasobu i wersja protokołu. Identyfikator zasobu pobierany jest z adresu URL. URL (*Uniform Resource Locator* – jednolity lokalizator zasobu) ma postać:

<schemat>:/<podmiot><ścieżka>/[?zapytanie][#fragment]

gdzie:

- ♦ <schemat> to http albo https,
- ♦ <podmiot> to nazwa lub IP hosta z opcjonalnym numerem portu i danymi uwierzytelniającymi,
- ♦ <ścieżka> to hierarchiczna ścieżka dostępu do zasobu, musi zaczynać się od znaku /,
- ♦ [zapytanie] jest opcjonalne i zawiera opcjonalne parametry.

Dla następującego adresu URL wpisanego do przeglądarki:
http://lukan.sytes.net:1880/test/get?parametr1=w1&parametr2=w2
początek żądania będzie miał postać:

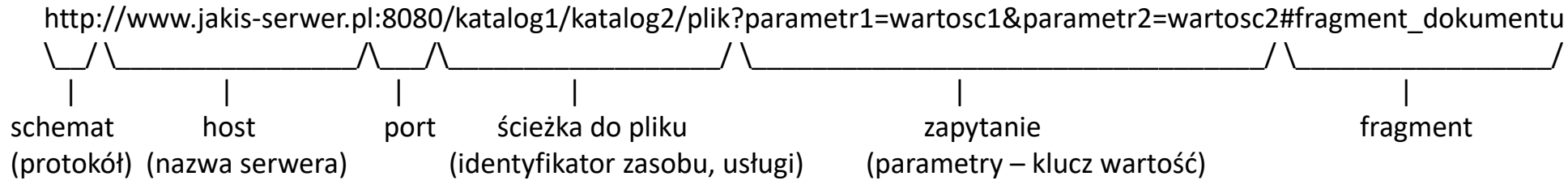
```
GET /test/get?parametr1=w1&parametr2=w2 HTTP/1.1
Host: lukan.sytes.net:1880
```

Druga linia żądania to obowiązkowy nagłówek Host. Po nagłówku obowiązkowym mogą pojawić się nagłówki opcjonalne.

Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Uniform Resource Identifier - URI

- *Ujednolicony Identyfikator Zasobów – identyfikuje zasoby w sieci*



Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Protokół HTTP Nagłówki

- Nagłówki są częścią składową żądania i odpowiedzi, zawierają informacje dla serwera i przeglądarki o parametrach żądania i odpowiedzi.
- Nagłówki między innymi informują drugą stronę w jakim formacie przesyłane są dane (treść).
- Standardowych nagłówków jest kilkadziesiąt, można dodawać własne niestandardowe nagłówki.
- Nagłówki są przesyłane jako kolekcja par nazwa-wartość.

Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Ważne nagłówki

Nazwa	Często stosowane wartości
Accept – informuje serwer o akceptowanych przez klienta typach dokumentu MIME, Content-Type – informuje o formacie MIME przesyłanej treści	text/plain text/html application/text; charset=utf-8 application/json application/xml application/soap+xml application/x-www-form-urlencoded
Accept-Encoding, Content-Encoding	gzip
Accept-Language, Content-Language	en, pl
Accept-Charset,	utf-8, iso-8859-1, iso-8859-2
Cookie	name=value; name2=value2; name3=value3
Allow – informuje o metodach HTTP obsługiwanych przez serwer	GET, POST, HEAD
If-Modified-Since – nakazuje przesłać dokument, jeśli został zmodyfikowany po podanej dacie	Data np.: 30 Oct 2019 20:00:00 GM

Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Odpowiedź HTTP

- Po otrzymaniu żądania serwer wykonuje zleconą akcję i wysyła odpowiedź. Akcją może być tylko przygotowanie odpowiedzi, co jest bardzo częste, gdy klient przegląda zasoby Internetu.
- W aplikacjach Internetu rzeczy akcją może być włączenie lub wyłączenie urządzeń, przekazanie do urządzeń komend sterujących, odczytanie stanu urządzeń lub wskazań przyrządów pomiarowych.
- W usługach internetowych akcją może być stworzenie na serwerze jakiegoś zasobu, jego modyfikacja lub usunięcie.
- Odpowiedzią najczęściej jest dokument, który można wyświetlić w przeglądarce, ale może to być także plik binarny, graficzny czy dane zapisane w dowolnym formacie. Odpowiedź, podobnie jak żądanie, składa się z nagłówków i treści. Nagłówki są podobne do nagłówków żądania.
- Odpowiedź zaczyna się od wersji protokołu HTTP, po którym następuje liczbowy kod statusu odpowiedzi i słowny opis statusu odpowiedzi np.:

HTTP/1.1 200 OK
HTTP/1.1 400 Bad Request

- Kod status odpowiedzi informuje aplikację klienta o statusie realizacji jego żądania.

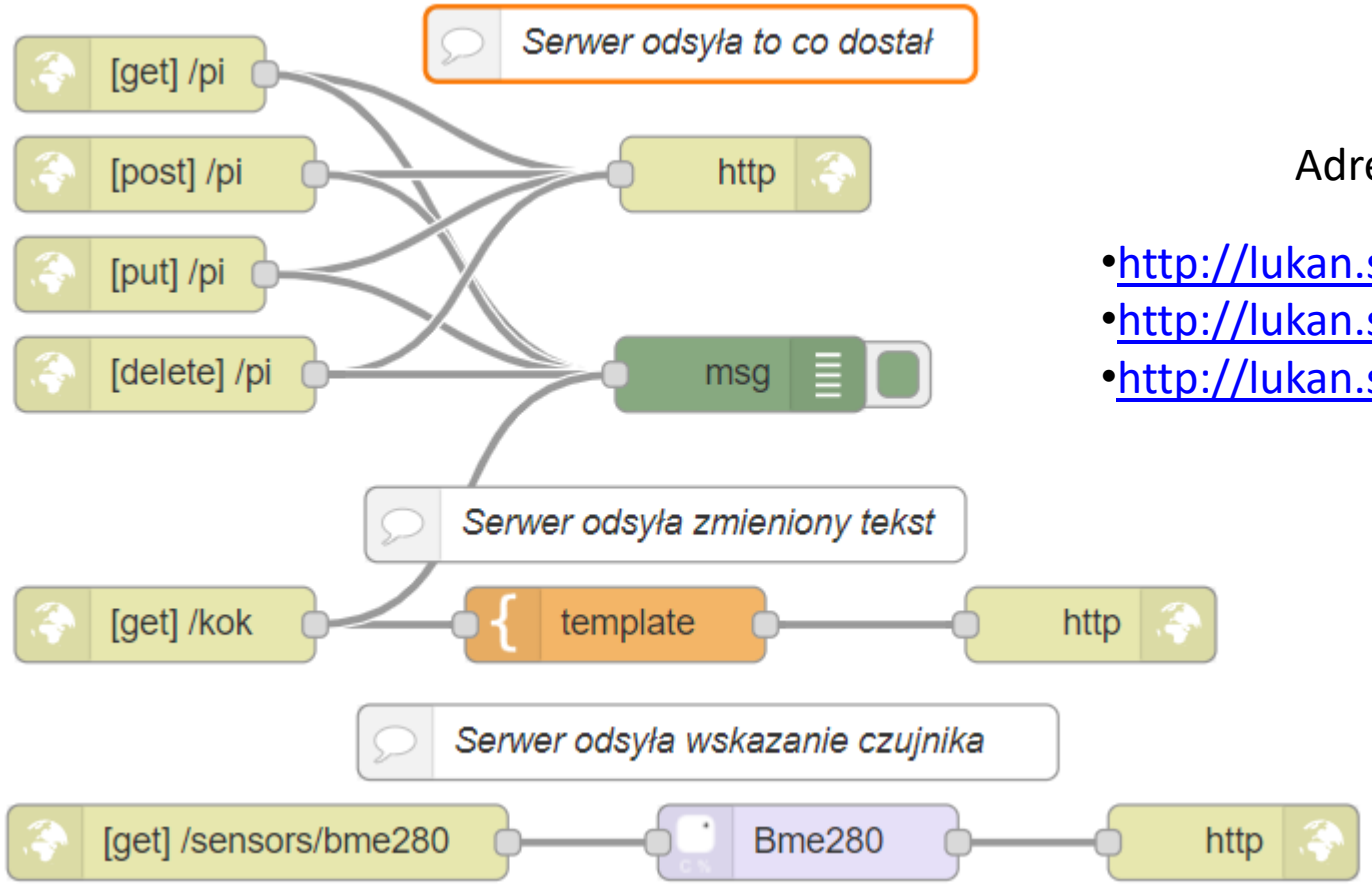
Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Często stosowane kody statusu odpowiedzi

Kod	Opis słowny	Znaczenie, zwrócony zasób
Kody informacyjne		
110	Connection Timed Out	Przekroczono czas połączenia, serwer zbyt długo nie odpowiada
111	Connection refused	Serwer odrzucił połączenie
Kody powodzenia		
200	OK	Zwrócono żądany dokument – najczęściej zwracany status
201	Created	Wysłany dokument został zapisany na serwerze
202	Accepted	Żądanie zostało przyjęte, ale jego realizacja jeszcze się nie skończyła
204	No content	Serwer zrealizował żądanie klienta i nie potrzebuje zwracać żadnej treści
Kody przekierowania		
304	Not Modified	Nie zmieniono zasobu, zawartość zasobu nie podległa zmianie według warunku przekazanego przez klienta (np. daty ostatniej wersji zasobu pobranej przez klienta i zapamiętanej w pamięć podręcznej przeglądarki)
Kody błędów po stronie klienta		
400	Bad Request	Żądanie nie może być obsłużone przez serwer z powodu nieprawidłowości postrzeganej jako błąd użytkownika
401	Unauthorized	Nieautoryzowany dostęp – realizacja żądania wymaga uwierzytelnienia
403	Forbidden	Serwer zrozumiał zapytanie, lecz konfiguracja bezpieczeństwa zabrania mu zwrócić żądany zasób
404	Not Found	Klient połączył się z serwerem, ale serwer nie może znaleźć żądanego zasobu – częsty błąd
Kody błędów po stronie serwera		
500	Internal Server Error	Wewnętrzny błąd serwera – serwer napotkał błąd, który uniemożliwił zrealizowanie żądania
501	Not Implemented	Nie zaimplementowano – serwer nie dysponuje funkcjonalnością określoną w żądaniu

Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Przykład lekkiego serwera zrealizowanego w Node-RED



Adresy

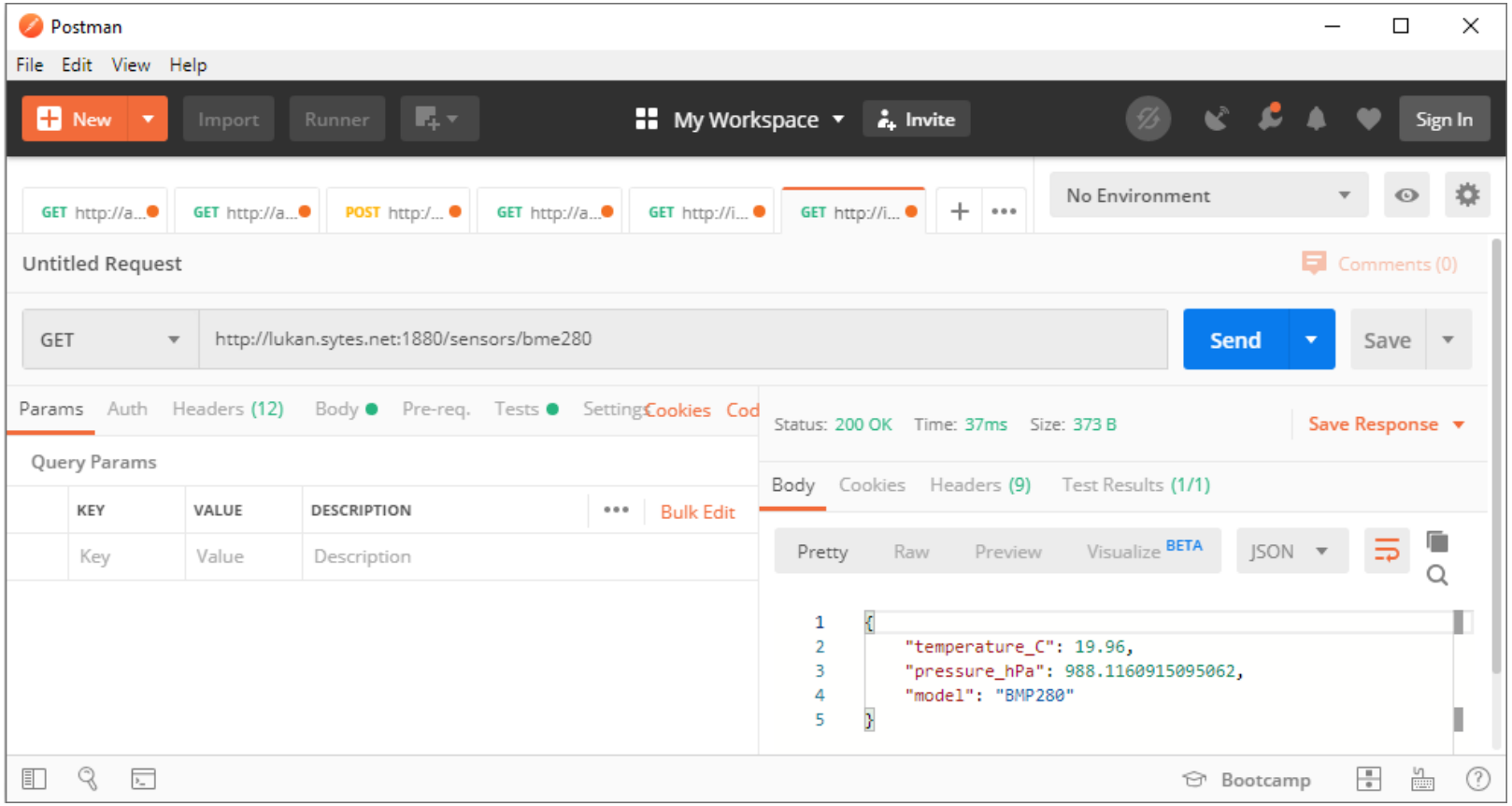
- http://lukan.sytes.net:1880/pi
- http://lukan.sytes.net:1880/kok
- http://lukan.sytes.net:1880/sensors/bme280

- Graf http://lukan.sytes.net:1880/#flow/95c9e167.8f0de

Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Postman

- Program do testowania usług korzystających z protokołu HTTP.
- Przeglądarka może wysyłać żądania GET z linii adresu, pozostałymi metodami z formularza na stronie.
- Program Postman wysyła żądania wieloma metodami z dowolną treścią i nagłówkami.

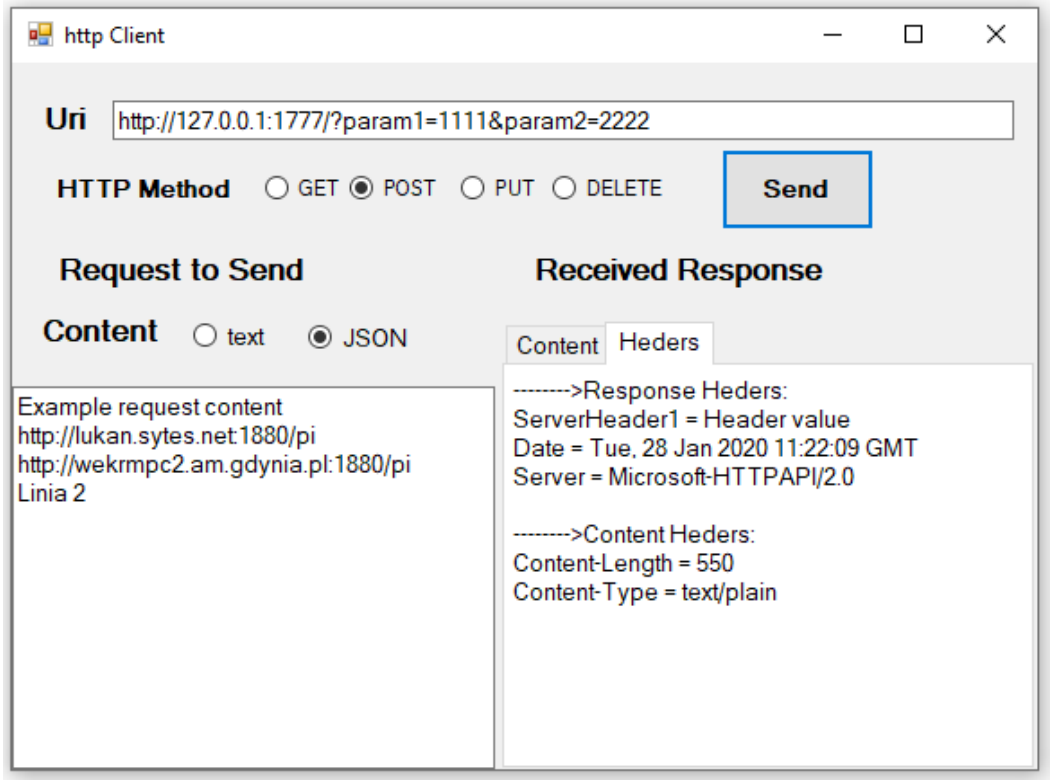


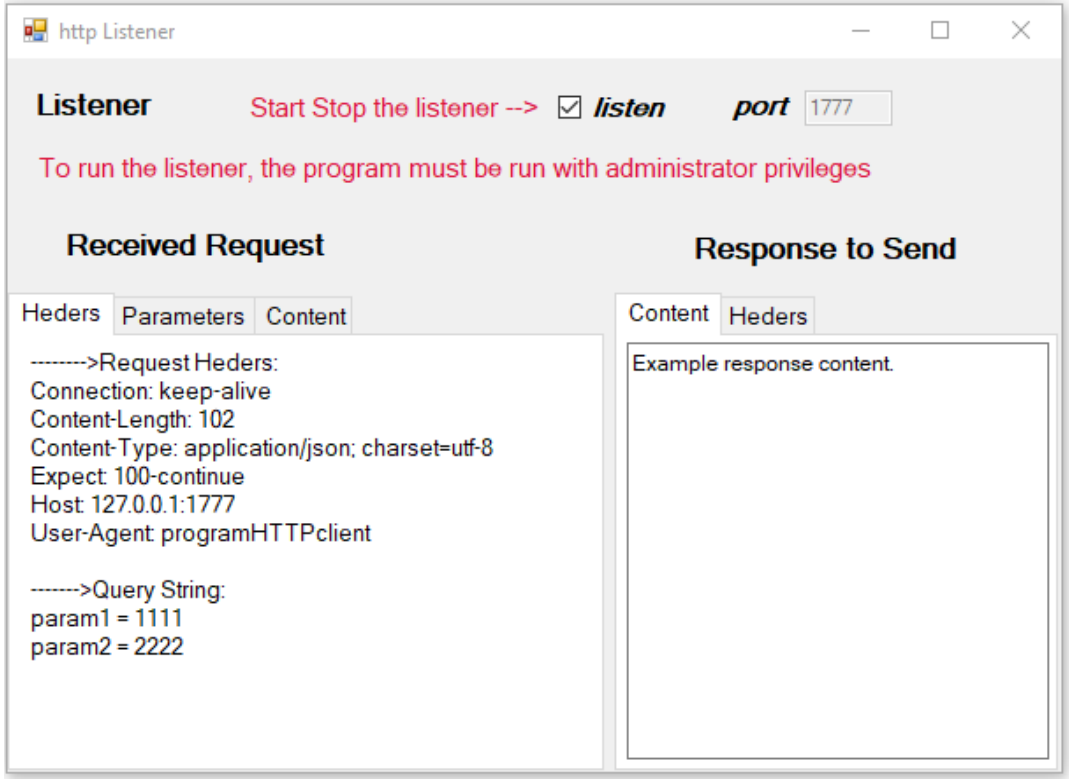
Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Programy klienta i serwera HTTP

Program: http://argo.umg.edu.pl/www/Internet/SendHttpReguest/
Kod: https://github.com/Andrzej1959/SendHttpRequest

Program: http://argo.umg.edu.pl/www/Internet/Listener/publikacja.htm
Kod: https://github.com/Andrzej1959/TestHttpListener





Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Message Queuing Telemetry Transport (MQTT)

- Protokół MQTT jest prostym i lekkim protokołem warstwy aplikacji.
- Protokół MQTT oparty o wzorzec publikacja/subskrypcja.
- Przeznaczony jest do transmisji dla urządzeń niewymagających dużej przepustowości.
- Poprzez ograniczenie prędkości transmisji, protokół zapewnia większą niezawodność.
- Protokół ten idealnie sprawdza się przy połączeniach maszyna-maszyna, w Internecie rzeczy, w urządzeniach mobilnych, oraz tam, gdzie wymagana jest oszczędność przepustowości, oraz energii.
- **Broker MQTT** pełni rolę serwera, z którym łączą się klienci (publikatorzy i subskrybenci), aby za jego pośrednictwem publikować i subskrybować wiadomości.
- Broker wymaga publicznego adresu IP, publikatorzy i subskrybenci nie wymagają.

Projekt „SezAM wiedzy, kompetencji i umiejętności" jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój