

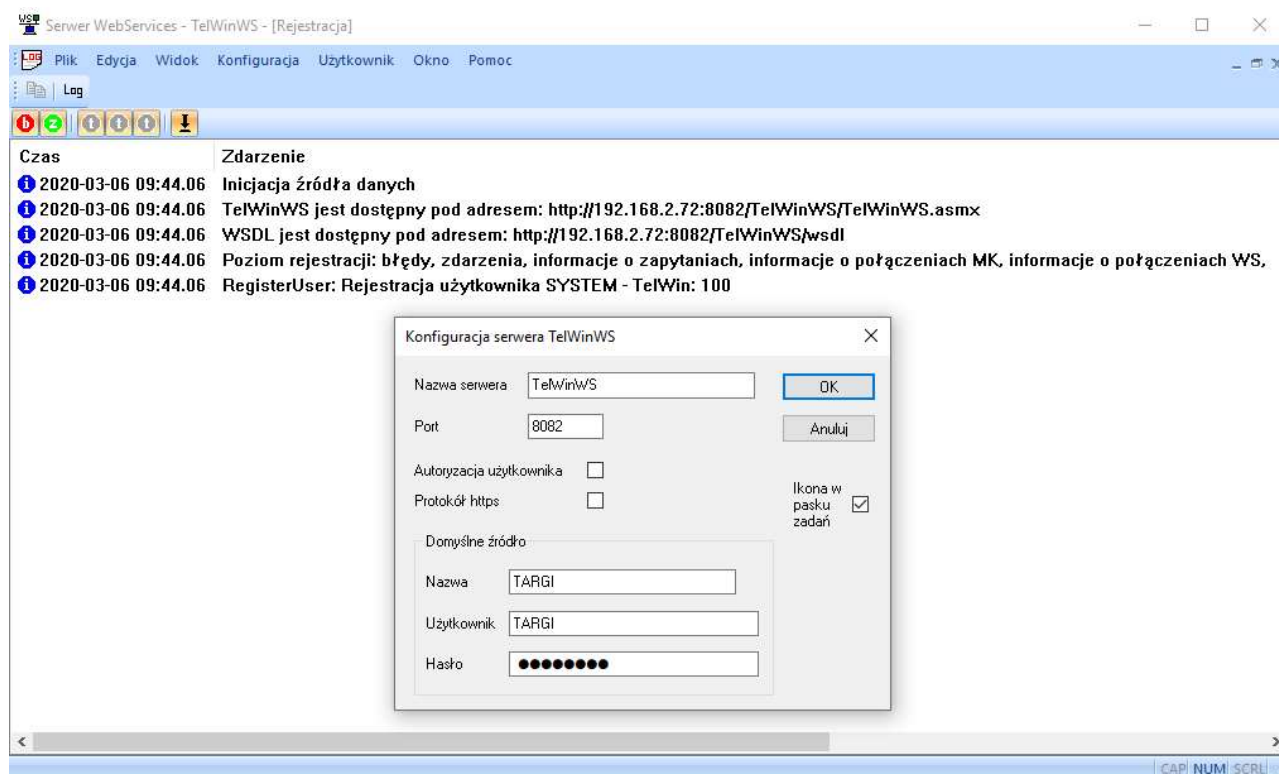
# TelWinWs

TelWinWs jest modułem, który umożliwia dostęp do danych systemu TelWin SCADA przy pomocy protokołu SOAP (technologia WebService). Dane w systemie TelWin SCADA widziane są za pośrednictwem tzw. zmiennych. Podstawowym atrybutem zmiennej jest jej nazwa, która jest unikalna. Zmienna reprezentuje następujące rodzaje danych:

- wartość bieżącą – aktualny pomiar z urządzenia
- wartości archiwalne – historia zmian wartości zmiennej z określonym kwantem czasowym
- wartości raportowe – wartości raportowe są zazwyczaj wyznaczone na podstawie wartości archiwalnych. Sposób agregowania danych zależy od definicji zmiennej (suma, średnia, wartość maksymalna, minimalna itp.). Okresem raportowym może być godzina, doba, miesiąc lub inny okres.

Moduł TelWinWs umożliwia odczyt i modyfikację wartości bieżących, archiwalnych i raportowych (godzinowych, dobowych i miesięcznych) oraz odczyt listy zmiennych.

Poniżej przedstawiono główne okno programu TelWinWs:



Dla każdego klienta może być uruchamiana osobna instancja TelWinWs. O uprawnieniach w systemie TelWin SCADA, które decydują do których zmiennych jest dostęp określonego typu, decyduje użytkownik skonfigurowany w programie TelWinWs. Każda instancja TelWinWs może kontrolować adresy, z których łączy się aplikacja kliencka. Lista dopuszczonych adresów konfigurowana jest pliku konfiguracyjnym TelWinWs.

### **Wydajność**

---

Należy unikać sytuacji odpytywania zmiennych pojedynczo. Najlepszy efekt uzyskuje się odpytując o większą liczę zmiennych (kilkaset) i większy okres czasu.

### Konfiguracja

Konfiguracja modułu TelWinWs jest określana za pomocą poleceń zawartych w menu *Konfiguracja*:

Konfiguracja serwera TelWinWS

Nazwa serwera: TelWinWS

Port: 8082

Autoryzacja użytkownika:

Protokół https:

Domyślne źródło

Nazwa: SZKOLENIE

Użytkownik: WS

Hasło: ●●●●●●●●

Ikona w pasku zadań:

OK, Anuluj

### Konfiguracja TelWinWs.ini

---

Szczegółowa konfiguracja modułu znajduje się w pliku TelWinWs.ini. Poniżej wyjaśnienie poszczególnych wpisów

### Sekcja Dostep

---

Sekcja odpowiada za definiowanie listy dostępu na podstawie adresów IP i maski sieci. Wpisy powinny mieć postać adresIP/maskaSieci=uprawnieniaRORW. Przykładowa sekcja poniżej

```
127.0.0.1/32=RW
192.168.1.78/32=RO
192.168.2.0/24=RW
```

### Sekcja Log

---

**Rodzaj=125**

---

Rodzaj logu

**Rozmiar=1000000**

---

Rozmiar pliku diagnostycznego w bajtach

### Sekcja Konfiguracja

---

#### **Nazwa=TelWinWs**

---

Nazwa modułu indentyfikowalna w module MK. Na podstawie nazwy należy utworzyć zmienne diagnostyczne w module TelSrv.

#### **WebPort=8082**

---

Port na którym interfejs ma być dostępny.

#### **Oczyszczanie=5**

---

#### **Adres=127.0.0.1**

---

Definicja dla jakiego adresu IP interfejs ma być dostępny.

#### **Autoryzacja uzytkownika=0**

---

•0 - autoryzacja wyłączona

•1 - autoryzacja użytkownika włączona

Aby zalogować użytkownika, należy wywołać funkcję *Login\_v2*, która zwróci id sesji. W kolejnych wywołaniach pozostałych funkcji (tylko w wersji v2) posługujemy się nadanym id sesji.

#### **Zrodlo=SZKOLENIE**

---

Źródło jest nazwą domyślnego modułu TelSrv/DataSrv, do którego ma być zalogowany interfejs. W przypadku wyłączonej autoryzacji zalogowanie następuje automatycznie po starcie serwera TelWinWs, gdy autoryzacja jest włączona, domyślne źródło służy do sprawdzenia poprawności wprowadzonych poświadczeń, podczas wywołania funkcji *Login\_v2*, następuje wtedy próba utworzenia połączenia do źródła i jest ono utrzymywane do czasu posiadania aktywnych sesji.

#### **Uzytkownik=WS**

---

Nazwa użytkownika systemu TelWin do połączenia z modułem TelSrv/DataSrv. Parametr jest niepotrzebny w przypadku włączonej opcji *Autoryzacja uzytkownika=1*

#### **Haslo=**

---

Przechowuje zakodowane hasło użytkownika. Parametr jest niepotrzebny w przypadku włączonej opcji *Autoryzacja uzytkownika=1*

#### **Https=0**

---

0 - komunikacja poprzez protokół http

1 - komunikacja poprzez zabezpieczony certyfikatem protokół https

Możemy posługiwać się certyfikatem, który sami wygenerujemy lub wydanym przez urząd certyfikacji. Jeśli posługujemy się certyfikatem wydanym przez urząd certyfikacji, należy powiązać certyfikat do adresu i portu, na którym wystawiamy TelWinWs'a.

**Instrukcja tworzenia własnego certyfikatu samopodpisanego (jeśli nie posiadamy certyfikatu wydanego przez urząd certyfikacji) i powiązania certyfikatu do adresu TelWinWs'a:**

•Certyfikat powinien zostać utworzony i zarejestrowany przed uruchomieniem aplikacji TelWinWs.

- Utworzenie certyfikatu instrukcją w PowerShell'u - tworzymy samodzielnie podpisany certyfikat dla lokalnej maszyny (ponieważ certyfikat musi być widoczny w systemie):

**New-SelfSignedCertificate -DnsName "127.0.0.1" -CertStoreLocation "cert:\LocalMachine\My"**

W kolejnym poleceniu użyć uzyskany "thumbprint" certyfikatu w opcji "certhash".

- Powiązanie certyfikatu do adresu i portu, na którym wystawiamy TelWinWs'a, robimy za pomocą komendy wywołanej w wierszu poleceń lub komendą w PowerShellu:

*UWAGA: hash certyfikatu można także sprawdzić z ustawień utworzonego certyfikatu, w tym celu wywołać okno "Uruchom", wpisać certlm.msc (certyfikaty lokalnej maszyny). W otwartym oknie przeglądania certyfikatów znaleźć nowo utworzony certyfikat rozwijając na drzewku gałąź "Osobiste"*

komenda do wywołania w wierszu poleceń cmd.exe:

**netsh http add sslcert ipport=127.0.0.1:8082  
certhash=4EA1A283CDA5E5490A493D250B86E40EEEA210ACappid={B9715467-2BE2-4408-B0A4-72E11CDD1549}**

komenda do wywołania w Windows PowerShell:

**netsh http add sslcert ipport=127.0.0.1:8082  
certhash=4EA1A283CDA5E5490A493D250B86E40EEEA210ACappid="{B9715467-2BE2-4408-B0A4-72E11CDD1549}"**

gdzie *ipport* to adres i port TelWinWs'a'. Jeśli posługujemy się certyfikatem podpisanym przez urząd certyfikacji, należy sprawdzić czy znajduje się on w osobistych certyfikatach lokalnej maszyny, jeśli nie należy go tam zaimportować.

- Dodać certyfikat samopodpisany do zaufanych. W tym celu należy wyeksportować utworzony certyfikat bez klucza prywatnego i zaimportować do gałęzi "Zaufane główne urzędy certyfikacji" oraz "Zaufani wydawcy"

- Zaimportować certyfikat samopodpisany z kluczem publicznym (bez klucza prywatnego) na maszynach klienckich.

## Sekcja Okna

---

### Ikona w pasku zadań= 1

---

Definiuje sposób zachowania okna aplikacji, czy będzie minimalizowana do paska zadań czy do ikony koło zegara.

### Funkcje dostępne na interfejsie SOAP

Funkcje są dostępne w dwóch wersjach. Uwaga, aby włączyć autoryzację należy w pliku INI podać Autoryzacja uzytkownika=1

Bez obsługi autoryzacji:

- [GetCurrentData](#)
- [GetHistoricalData](#)
- [WriteData](#)
- [GetVariablesList](#)
- [GetInterfaceInfo](#)

Z obsługą autoryzacji:

- [Login\\_v2](#)
- [Logout\\_v2](#)
- [GetCurrentData\\_v2](#)
- [GetHistoricalData\\_v2](#)
- [WriteData\\_v2](#)
- [GetVariablesList\\_v2](#)

### Przekazywanie nazw zmiennych w funkcjach interfejsowych

---

Sposoby:

- **zmienna** - podając nazwę zmiennej. Zmienna zostanie pobrana z domyślnego źródła podanego w konfiguracji serwera TelWinWs
- **źródło@zmienna** - podając nazwę zmiennej i źródło, w którym dana zmienna się znajduje.

W pojedynczym wywołaniu funkcji interfejsowej można łączyć obydwie zapisy i wydawać zapytanie dotyczące zmiennych znajdujących się w paru źródłach.

### GetCurrentData – umożliwia odczyt wartości bieżących zmiennych

---

Parametry:

- **long** idSession – parametr używany informacyjnie do logowania konkretnych sekwencji zapytania
- **string []** varNames – tablica elementów typu string, w których znajdują nazwy zmiennych. Patrz [przekazywanie nazw zmiennych w funkcjach interfejsowych](#)

Zwraca:

- **WsData []** GetCurrentDataResult – lista elementów WsData. Patrz opis struktury [WsData](#)

# TelWin SCADA - dokumentacja systemu

Przykład wywołania funkcji w narzędziu SoapUI. Pobranie jednej zmiennej:

The screenshot displays the SoapUI interface for a SOAP request and response. The request is a `GetCurrentData` call with a session ID and a variable name `SYM01`. The response is a `GetCurrentDataResponse` containing a single `WsData` element for `SYM01` with a value of `171`.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tel="http://www.tel-ster.pl/TelWinWS">
  <soapenv:Header/>
  <soapenv:Body>
    <tel:GetCurrentData>
      <tel:idSession>1</tel:idSession>
      <tel:varNames>
        <tel:string>SYM01</tel:string>
      </tel:varNames>
    </tel:GetCurrentData>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  <s:Body>
    <GetCurrentDataResponse xmlns="http://www.tel-ster.pl/TelWinWS">
      <GetCurrentDataResult>
        <WsData>
          <VarName>SYM01</VarName>
          <DataType>0</DataType>
          <Values>
            <WsVal>
              <ValType>171</ValType>
              <ValFlo>15.02685546875</ValFlo>
              <ValStr/>
            </WsVal>
          </Values>
        </WsData>
      </GetCurrentDataResult>
    </GetCurrentDataResponse>
  </s:Body>
</s:Envelope>
```

response time: 192ms (451 bytes)

Przykład wywołania funkcji w narzędziu SoapUI. Pobranie wielu zmiennych:

The screenshot displays the SoapUI interface for a SOAP request and response. The request is a `GetCurrentData` call with a session ID and three variable names: `SYM01`, `SYM02`, and `SYM03`. The response is a `GetCurrentDataResponse` containing three `WsData` elements for `SYM01`, `SYM02`, and `SYM03`, each with its respective value.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tel="http://www.tel-ster.pl/TelWinWS">
  <soapenv:Header/>
  <soapenv:Body>
    <tel:GetCurrentData>
      <tel:idSession>1</tel:idSession>
      <tel:varNames>
        <tel:string>SYM01</tel:string>
        <tel:string>SYM02</tel:string>
        <tel:string>SYM03</tel:string>
      </tel:varNames>
    </tel:GetCurrentData>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  <s:Body>
    <GetCurrentDataResponse xmlns="http://www.tel-ster.pl/TelWinWS">
      <GetCurrentDataResult>
        <WsData>
          <VarName>SYM01</VarName>
          <DataType>0</DataType>
          <Values>
            <WsVal>
              <ValType>171</ValType>
              <ValFlo>97.5816455078125</ValFlo>
              <ValStr/>
            </WsVal>
          </Values>
        </WsData>
        <WsData>
          <VarName>SYM02</VarName>
          <DataType>0</DataType>
          <Values>
            <WsVal>
              <ValType>171</ValType>
              <ValFlo>88.6810302734375</ValFlo>
              <ValStr/>
            </WsVal>
          </Values>
        </WsData>
        <WsData>
          <VarName>SYM03</VarName>
          <DataType>0</DataType>
          <Values>
            <WsVal>
              <ValType>171</ValType>
              <ValFlo>67.3095703125</ValFlo>
              <ValStr/>
            </WsVal>
          </Values>
        </WsData>
      </GetCurrentDataResult>
    </GetCurrentDataResponse>
  </s:Body>
</s:Envelope>
```

response time: 142ms (898 bytes)

## GetHistoricalData – umożliwiają odczyt danych archiwalnych i raportowych

Parametry:

- **long** idSession – parametr używany informacyjnie do logowania konkretnych sekwencji zapytania
- **int** dataType – typ danych jakiego dotyczy zapytanie:
  - 1 - archiwalne
  - 2 - godzinowe
  - 4 - dobowe
  - 8 - miesięczne
- **DateTime** dateStart - początek okresu jakiego dotyczy polecenie
- **DateTime** dateEnd – koniec okresu jakiego dotyczy polecenie
- **string []** varNames – tablica obiektów typu string, w których znajdują nazwy zmiennych, dla których mają być odczytane wartości. Patrz [przekazywanie nazw zmiennych w funkcjach interfejsowych](#)

Zwraca:

- **WsData []** GetHistoricalDataResult – tablica obiektów WsData, zawierająca żądane wartości. Patrz opis struktury [WsData](#)

Przykład wywołania funkcji w narzędziu SoapUI. Pobranie danych archiwalnych:

The screenshot displays the SoapUI interface with a SOAP request on the left and its corresponding response on the right. The request is a SOAP envelope containing a `GetHistoricalData` operation with parameters: `idSession`, `dataType`, `dateStart`, `dateEnd`, and `varNames`. The response is a SOAP envelope containing a `GetHistoricalDataResponse` with a `GetHistoricalDataResult` array of `WsData` objects. Each `WsData` object includes `VarName`, `DataType`, and `Values` (a `WsVal` array). The `WsVal` array contains `ValType`, `ValFlo`, and `ValStr` elements, along with `Status` and `TStamp` information.

```
<?xml version='1.0' encoding='utf-8'>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <x:tel xmlns:tel='http://www.tel-ster.pl/TelWinWS/'>
    <soap:Header/>
    <soap:Body>
      <tel:GetHistoricalData>
        <tel:idSession>1</tel:idSession>
        <tel:dataType>1</tel:dataType>
        <tel:dateStart>2018-07-19T09:00:00</tel:dateStart>
        <tel:dateEnd>2018-07-19T10:00:00</tel:dateEnd>
        <tel:varNames>
          <tel:string>SYM01</tel:string>
        </tel:varNames>
      </tel:GetHistoricalData>
    </soap:Body>
  </x:tel>
</soap:Envelope>
```

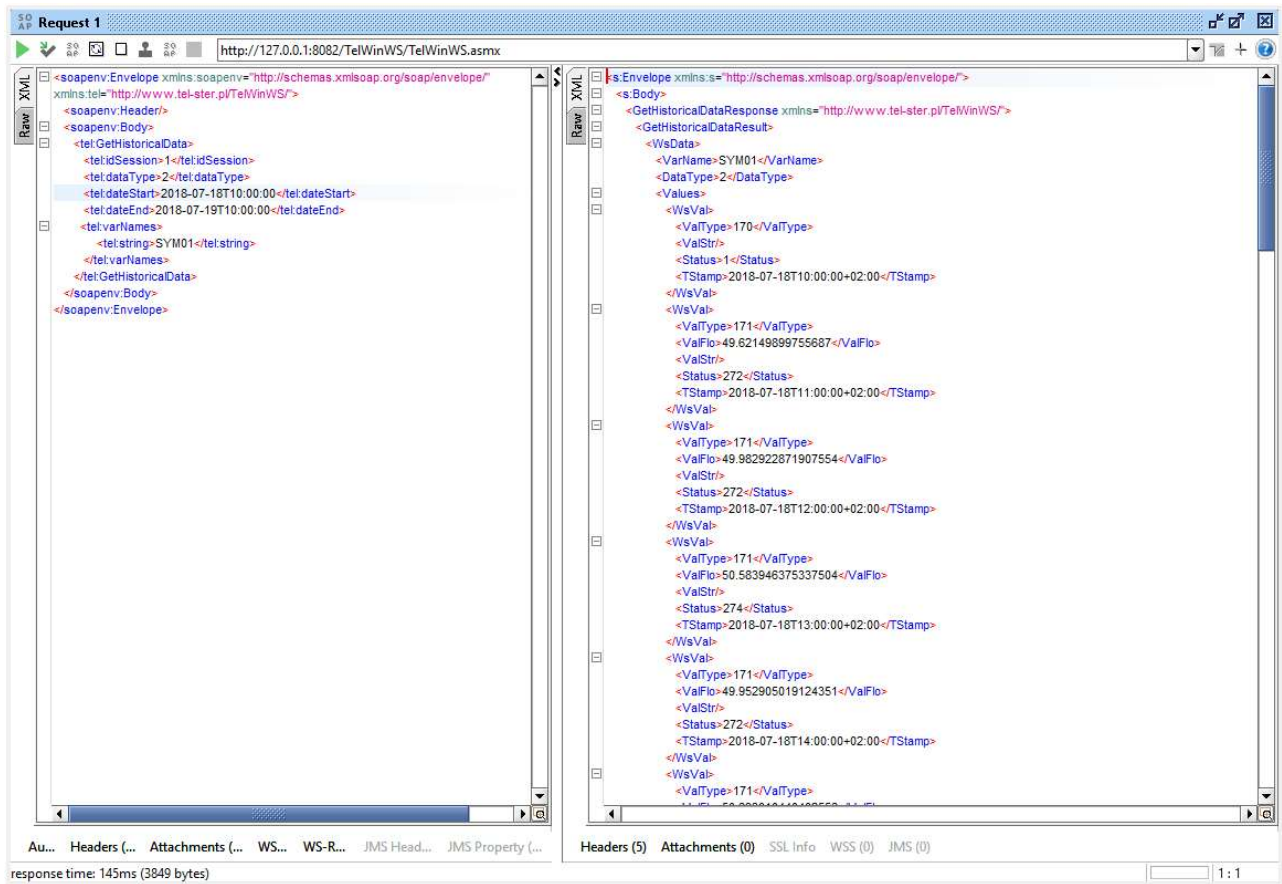
```
<?xml version='1.0' encoding='utf-8'>
<s:Envelope xmlns:s='http://schemas.xmlsoap.org/soap/envelope/'>
  <s:Body>
    <GetHistoricalDataResponse xmlns='http://www.tel-ster.pl/TelWinWS/'>
      <GetHistoricalDataResult>
        <WsData>
          <VarName>SYM01</VarName>
          <DataType>1</DataType>
          <Values>
            <WsVal>
              <ValType>171</ValType>
              <ValFlo>71.6033935546875</ValFlo>
              <ValStr/>
              <Status>16</Status>
              <TStamp>2018-07-19T08:59:59.511+02:00</TStamp>
            </WsVal>
            <WsVal>
              <ValType>171</ValType>
              <ValFlo>28.717041015625</ValFlo>
              <ValStr/>
              <Status>16</Status>
              <TStamp>2018-07-19T09:00:00.011+02:00</TStamp>
            </WsVal>
            <WsVal>
              <ValType>171</ValType>
              <ValFlo>84.8052978515625</ValFlo>
              <ValStr/>
              <Status>16</Status>
              <TStamp>2018-07-19T09:00:00.511+02:00</TStamp>
            </WsVal>
            <WsVal>
              <ValType>171</ValType>
              <ValFlo>85.2203389140625</ValFlo>
              <ValStr/>
              <Status>16</Status>
              <TStamp>2018-07-19T09:00:01.011+02:00</TStamp>
            </WsVal>
            <WsVal>
              <ValType>171</ValType>
              <ValFlo>44.342041015625</ValFlo>
              <ValStr/>
              <Status>16</Status>
              <TStamp>2018-07-19T09:00:01.511+02:00</TStamp>
            </WsVal>
          </Values>
        </WsData>
      </GetHistoricalDataResult>
    </s:Body>
  </s:Envelope>
```

response time: 436ms (1029335 bytes) Headers (5) Attachments (0) SSL info WSS (0) JMS (0) 2 : 1



# TelWin SCADA - dokumentacja systemu

Przykład wywołania funkcji w narzędziu SoapUI. Pobranie raportowych godzinowych:



# TelWin SCADA - dokumentacja systemu

Przykład wywołania funkcji w narzędziu SoapUI. Pobranie raportowych dobowych:

The screenshot displays the SoapUI interface for a SOAP request and response. The URL is `http://127.0.0.1:8082/TelWinWS/TelWinWS.asmx`. The request is a SOAP envelope with the following structure:

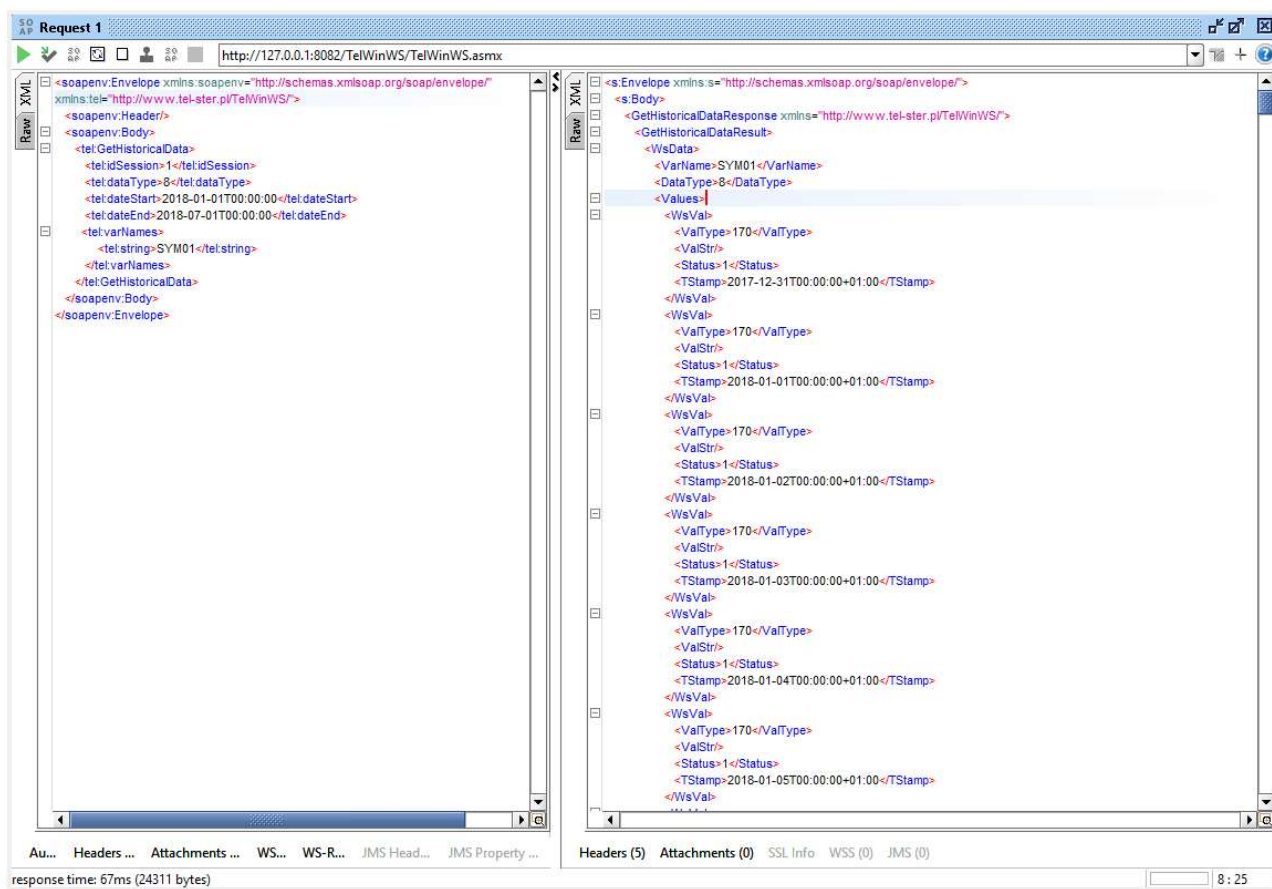
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tel="http://www.tel-ster.pl/TelWinWS">
  <soapenv:Header/>
  <soapenv:Body>
    <tel:GetHistoricalData>
      <tel:idSession>1</tel:idSession>
      <tel:dataType>4</tel:dataType>
      <tel:dateStart>2018-07-10T10:00:00</tel:dateStart>
      <tel:dateEnd>2018-07-19T10:00:00</tel:dateEnd>
      <tel:varNames>
        <tel:string>SYM01</tel:string>
      </tel:varNames>
    </tel:GetHistoricalData>
  </soapenv:Body>
</soapenv:Envelope>
```

The response is a SOAP envelope with the following structure:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetHistoricalDataResponse xmlns="http://www.tel-ster.pl/TelWinWS">
      <GetHistoricalDataResult>
        <WsData>
          <VarName>SYM01</VarName>
          <DataType>4</DataType>
          <Values>
            <WsVal>
              <ValType>171</ValType>
              <ValFlo>50.078177858282011</ValFlo>
              <ValStr/>
              <Status>272</Status>
              <TStamp>2018-07-10T00:00:00+02:00</TStamp>
            </WsVal>
            <WsVal>
              <ValType>171</ValType>
              <ValFlo>49.991716014014351</ValFlo>
              <ValStr/>
              <Status>272</Status>
              <TStamp>2018-07-11T00:00:00+02:00</TStamp>
            </WsVal>
            <WsVal>
              <ValType>171</ValType>
              <ValFlo>50.017741998036691</ValFlo>
              <ValStr/>
              <Status>272</Status>
              <TStamp>2018-07-12T00:00:00+02:00</TStamp>
            </WsVal>
            <WsVal>
              <ValType>171</ValType>
              <ValFlo>49.936212134152328</ValFlo>
              <ValStr/>
              <Status>272</Status>
              <TStamp>2018-07-13T00:00:00+02:00</TStamp>
            </WsVal>
            <WsVal>
              <ValType>171</ValType>
              <ValFlo>49.983642118948474</ValFlo>
              <ValStr/>
              <Status>272</Status>
              <TStamp>2018-07-14T00:00:00+02:00</TStamp>
            </WsVal>
          </Values>
        </WsData>
      </GetHistoricalDataResult>
    </GetHistoricalDataResponse>
  </s:Body>
</s:Envelope>
```

The status bar at the bottom shows: Headers (5) Attachments (0) SSL Info WSS (0) JMS (0) response time: 176ms (1746 bytes) 1:1

Przykład wywołania funkcji w narzędziu SoapUI. Pobranie raportowych miesięcznych:



### WriteData – umożliwia modyfikację wartości bieżących i raportowych

Parametry:

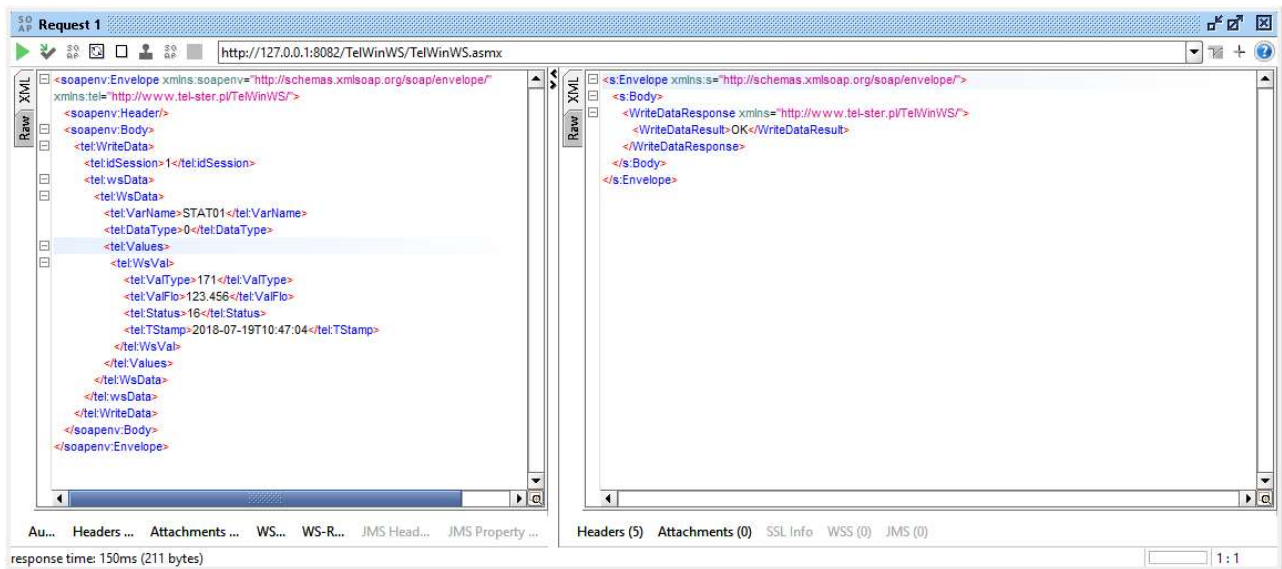
- **long** idSession – parametr używany informacyjnie do logowania konkretnych sekwencji zapytania
- **WsData[]** wsData – tablica obiektów WsData zawierająca wartości, które mają być zmodyfikowane. Patrz opis struktury [WsData](#)

Zwraca:

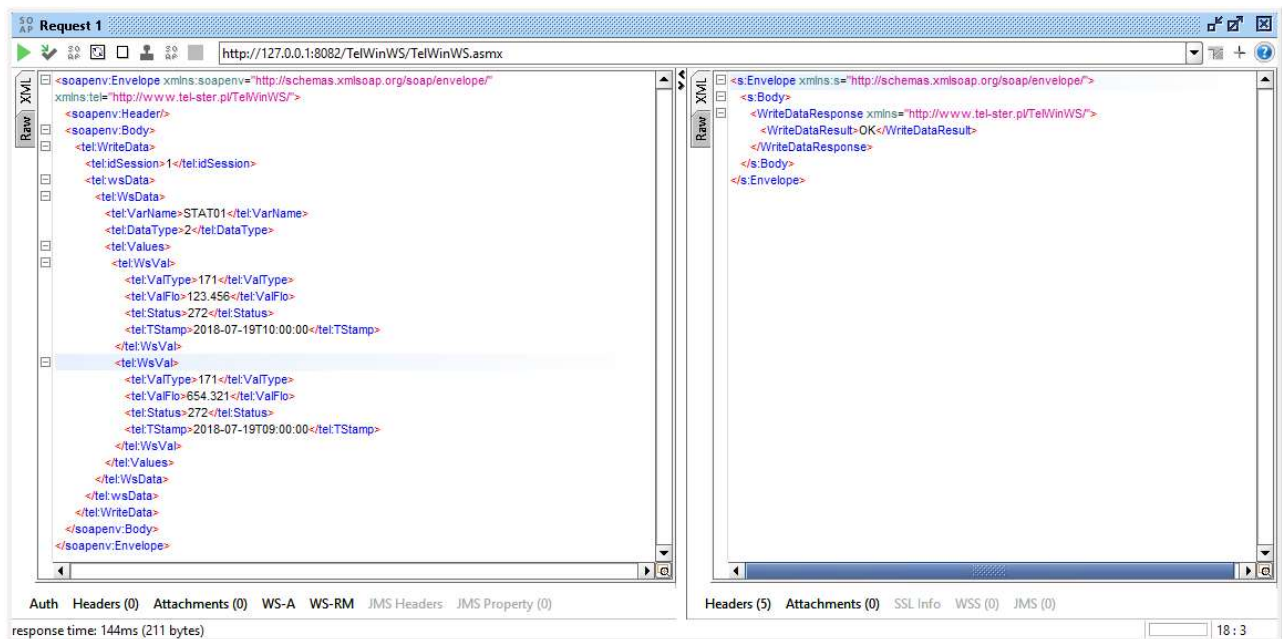
- **string** WriteDataResult – informacja o wyniku modyfikacji wartości

# TelWin SCADA - dokumentacja systemu

Przykład wywołania funkcji w narzędziu SoapUI. Modyfikacja wartości bieżącej:



Przykład wywołania funkcji w narzędziu SoapUI. Modyfikacja wartości raportowej godzinowej:



## GetVariablesList – umożliwia odczyt listy zmiennych

Parametry:

- **long** idSession – parametr używany informacyjnie do logowania konkretnych sekwencji zapytania
- **string** varFilter – wzorzec filtrujący nazwy zmiennych, które będą pobrane

Wzorzec filtrujący możemy przekazać na dwa sposoby:

- źródło@filtr - wpisując konkretne źródło, z którego pobieramy listę zmiennych, np. TARGI@\*, gdzie TARGI to nazwa źródła, @ to znak rozdzielający nazwę źródła od filtru "\*"

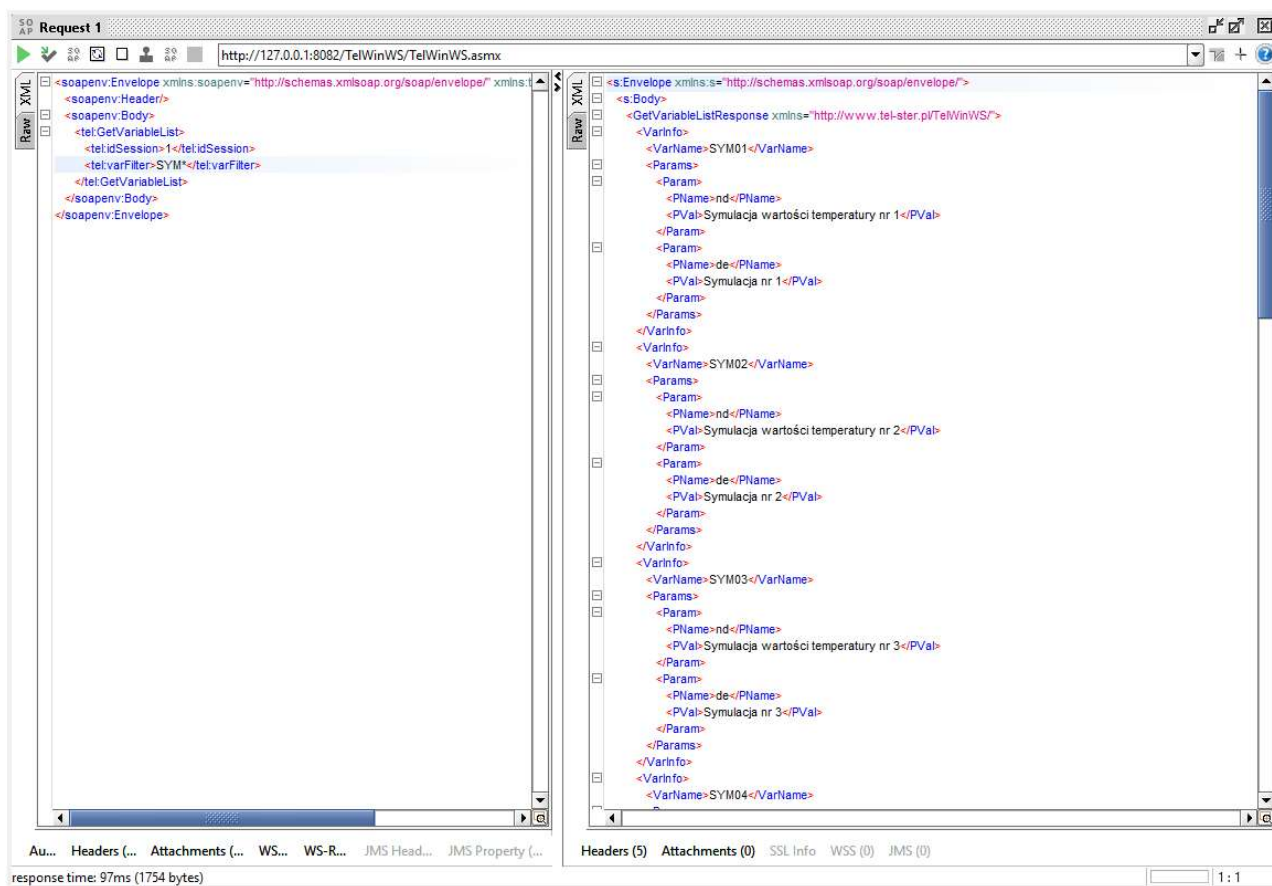
## TelWin SCADA - dokumentacja systemu

o filtr - wpisując jedynie filtr, pobrana zostanie wtedy lista zmiennych z domyślnego źródła podanego w konfiguracji serwera TelWinWs.

Zwraca:

• **VarDefInfo[]** GetVariableListResult – tablica obiektów VarDefInfo. Patrz opis struktury [VarDefInfo](#)

Przykład wywołania funkcji w narzędziu SoapUI:



## GetInterfaceInfo - umożliwia odczyt informacji o wersji modułu

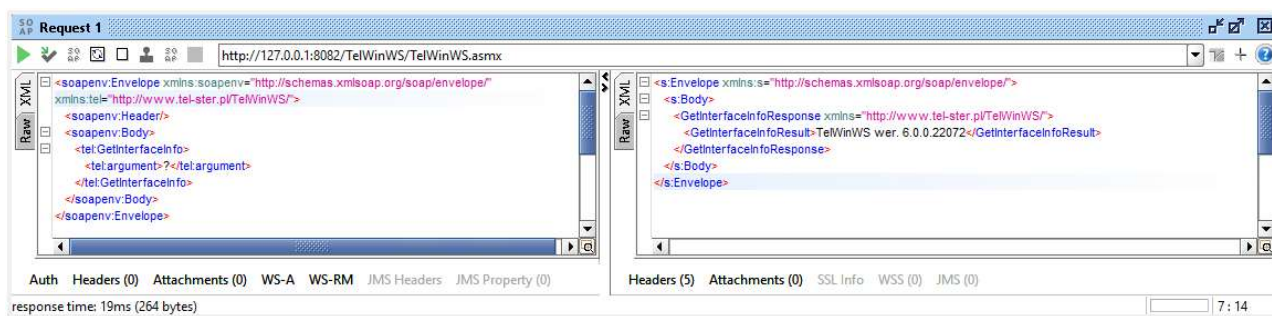
Parametry:

• **string** argument – parametr obecnie nieużywany, należy podać dowolny string

Zwraca:

• **string** GetInterfaceInfo – wersja modułu TelWinWs

Przykład wywołania funkcji w narzędziu SoapUI:



---

**Login\_v2 – umożliwia autoryzację użytkownika, przydzielony id sesji jest ważny przez 5 minut, kolejne wywołania pozostałych funkcji, obsługujących autoryzację (v2), powodują odświeżenie czasu ponownie do 5 minut.**

---

Parametry:

- string []** user – nazwa użytkownika
- string []** password – hasło użytkownika

Zwraca:

- string []** Login\_v2Response – idSession - guid będący unikalnym id sesji, który jest niezbędny do kolejnych wywołań pozostałych funkcji v2

---

**Logout\_v2 – umożliwia wylogowanie sesji użytkownika (wersja z autoryzacją)**

---

Parametry:

- string []** idSession – unikalny guid sesji, używany do autoryzacji sesji zalogowanego użytkownika

---

**GetCurrentData\_v2 – umożliwia odczyt wartości bieżących zmiennych (wersja z autoryzacją)**

---

Parametry:

- string []** idSession – unikalny guid sesji, używany do autoryzacji sesji zalogowanego użytkownika
- string []** varNames – tablica elementów typu string, w których znajdują nazwy zmiennych. Patrz [przekazywanie nazw zmiennych w funkcjach interfejsowych](#)

Zwraca:

- WsData []** GetCurrentData\_v2Result – lista elementów WsData - opis elementu WsData poniżej. Patrz opis struktury [WsData](#)



### GetHistoricalData\_v2 – umożliwia odczyt danych archiwalnych i raportowych (wersja z autoryzacją)

---

Parametry:

- string []** idSession – unikalny guid sesji, używany do autoryzacji sesji zalogowanego użytkownika
- int** dataType – typ danych jakiego dotyczy zapytanie:
  - 1 - archiwalne
  - 2 - godzinowe
  - 4 - dobowe
  - 8 - miesięczne
- DateTime** dateStart – początek okresu jakiego dotyczy polecenie
- DateTime** dateEnd – koniec okresu jakiego dotyczy polecenie
- string []** varNames – tablica obiektów typu string, w których znajdują nazwy zmiennych, dla których mają być odczytane wartości. Patrz [przekazywanie nazw zmiennych w funkcjach interfejsowych](#)

Zwraca:

- WsData []** GetHistoricalData\_v2Result – tablica obiektów WsData, zawierająca żądane wartości. Patrz opis struktury [WsData](#)

### WriteData\_v2 – umożliwia modyfikację wartości bieżących i raportowych (wersja z autoryzacją)

---

Parametry:

- string []** idSession – unikalny guid sesji, używany do autoryzacji sesji zalogowanego użytkownika
- WsData[]** wsData – tablica obiektów WsData zawierająca wartości, które mają być zmodyfikowane. Patrz opis struktury [WsData](#)

Zwraca:

- string** WriteDataResult – informacja o wyniku modyfikacji wartości

### GetVariablesList\_v2 – umożliwia odczyt listy zmiennych (wersja z autoryzacją)

---

Parametry:

- string []** idSession – unikalny guid sesji, używany do autoryzacji sesji zalogowanego użytkownika
- string** varFilter – wzorzec filtrujący nazwy zmiennych, które będą pobrane

Wzorzec filtrujący możemy przekazać na dwa sposoby:

- źródło@filtr - wpisując konkretne źródło, z którego pobieramy listę zmiennych, np. TARGI@\*, gdzie TARGI to nazwa źródła, @ to znak rozdzielający nazwę źródła od filtru "\*"

o filtr - wpisując jedynie filtr, pobrana zostanie wtedy lista zmiennych z domyślnego źródła podanego w konfiguracji serwera TelWinWs.

Zwraca:

• **VarDefInfo[]** GetVariableList\_v2Result – tablica obiektów VarDefInfo. Patrz opis struktury [VarDefInfo](#)



### Struktura obiektu WsData

Obiekt **WsData** służy do reprezentacji zbioru wartości danego parametru, identyfikowalnego za pomocą nazwy i składa się z następujących elementów:

•**string** VarName – nazwa parametru. Patrz [przekazywanie nazw zmiennych w funkcjach interfejsowych](#)

•**int** DataType – typ wartości w elemencie Values:

- 0 - wartości bieżące
- 1 - wartości archiwalne
- 2 - wartości godzinowe
- 4 - wartości dobowe
- 8 - wartości miesięczne

•**int** DataStatus – informacja o statusie pozyskania wartości

- 0 - wartość poprawna - PTDT\_OK
- 1 - brak zmiennej - PTDT\_NO\_VARIABLE\_ERROR
- 2 - brak dostępu do zmiennej - PTDT\_NO\_ACCESS\_ERROR
- 3 - brak połączenia - PTDT\_NO\_CONNECTION\_ERROR
- 4 - nieprawidłowe dane, brak połączenia - PTDT\_DATA\_ERROR
- 5 - błąd modyfikacji wartości - PTDT\_MODIFY\_ERROR
- 6 - błąd ustawienia czasu - TIME\_SET\_ERROR
- 7 - błędny czas - PTDT\_TIME\_ERROR

•**WsVal[]** Values - tablica odpowiednich wartości parametru. Patrz opis struktury [WsVal](#)

### Struktura obiektu WsVal

Obiekt **WsVal** służy do reprezentacji pojedynczej wartości pomiarowej i składa się z następujących elementów:

- **int** ValType – typ wartości (zgodnie z definicją w TelWin SCADA):
  - anyVal = 100
  - boolVal = 120
  - int8Val = 131
  - uint8Val = 132
  - int16Val = 140
  - uint16Val = 141
  - int32Val = 150
  - uint32Val = 151
  - int64Val = 160
  - uint64Val = 161
  - float4Val = 170
  - float8Val = 171
  - float10Val = 172
  - strVal = 190
  - badVal = 255
- **long** ValInt – wartość jako liczba całkowita
- **double** ValFlo - wartość jako liczba zmiennoprzecinkowa
- **string** ValStr – wartość jako string
- **int** Status – pole bitowe określające status wartości (zgodnie z definicją w TelWin SCADA):
  - 1 - wartość nieznana
  - 2 – błąd transmisji
  - 4 – alarm aktywny
  - 8 – alarm potwierdzony
  - 16 – wartość pomiarowa,
  - 32 – nastawa ręczna
  - 64 – błąd obliczenia
  - 128 – raport pobrany
  - 256 – raport zainicjowany
  - 512 – blokada
  - 16384 – pamięć alarmu
- **DateTime** TStamp - etykieta czasowa wartości

Na podstawie wartości w polu ValType należy korzystać w właściwego pola ValInt, ValFlo lub ValStr.

### Struktura obiektu VarDefInfo

Obiekt **VarDefInfo** służy do reprezentacji zmiennej wraz z jej parametrami i ich wartościami. Składa się z następujących elementów:

- string** VarName – nazwa zmiennej
- VarDefInfoParam[]** Params – lista parametrów wraz z ich wartościami.

Poniżej opis struktury VarDefInfoParam:

- string** PName – nazwa parametru
- string** PVal – wartość parametru

Lista dostępnych parametrów zmiennych PName:

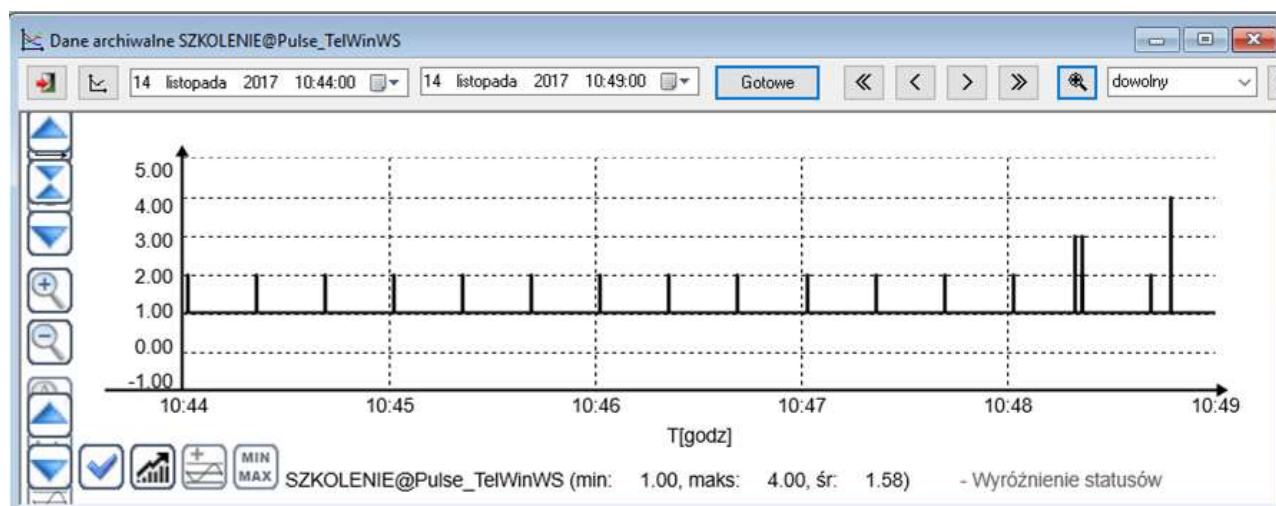
- de** - Opis zmiennej
- nd** - Nazwa pełna zmiennej

### Monitorowanie modułu TelWinWs

Moduł TelWinWs pozwala monitorować status swojej pracy. W tym celu należy w serwerze systemu utworzyć trzy zmienne o nazwach:

- **ReadNum\_TelWinWs** - zmienna przechowuje liczbę odczytanych wartości
- **WriteNum\_TelWinWs** - zmienna przechowuje liczbę zapisanych wartości
- **Pulse\_TelWinWs** - zmienna przechowuje puls życia aplikacji:
  - 1 - puls życia
  - 2 - puls życia
  - 3 - otrzymano zlecenie odczytu danych
  - 4 - otrzymano zlecenie modyfikacji danych
  - 5 - otrzymano zlecenie pobrania listy zmiennych

Gdzie część **TelWinWs** zastępujemy nazwą serwera z konfiguracji.



### Monitorowanie działań użytkowników

Logi aplikacji zapisywane są w pliku TelWinWs.log. W pliku tym, nie są zapisywane o modyfikowanych danych. O tym co zmodyfikowali klienci dowiemy się z logu TelSrv\_a\_ws.log.

2017-11-14 11:23.48 ADMIN z TelWinWs: Modyfikacja wartości bieżącej - zmienna SYM01 (v: 78.997802734375 -> 605, s: 16 -> 16)

2017-11-14 11:22.48 ADMIN z TelWinWs: Modyfikacja wartości archiwalnych - zmienna SYM01 (2017-11-13 00:00 <-> 2017-11-13 23:59, #1440)

2017-11-14 10:48.47 ADMIN z TelWinWs: Modyfikacja wartości raportowej godzinowej - zmienna SYM01 2017-11-13 00:00.00 (v: 49.7256028917101 -> 12, s: 272 -> 272)

2017-11-14 11:19.45 ADMIN z TelWinWs: Modyfikacja wartości raportowej dobowej - zmienna SYM01 2017-11-13 00:00.00 (v: 39.6452671333596 -> 85, s: 272 -> 272)

### Pliki diagnostyczne

#### **TelWinWs.log**

---

W zależności od wybranego poziomu rejestracji, plik ten zawiera wpisy dotyczące:

- wykrytych błędów,
- zdarzeń,
- zapytań,
- połączeń MK,
- połączeń WS

#### **TelWinWs\_d.log**

---

Ogólny plik diagnostyczny modułu.