

Funkcja reguł CallWebservice

1. Zastosowanie

Funkcja umożliwia wysyłanie z poziomu AMODIT żądań do serwisów internetowych wspierających protokół SOAP. Inną podobną funkcją wspieraną w regułach jest CallRest, z tym, że funkcja ta wspiera standard REST.

2. Wywoływanie funkcji

Funkcję należy wywoływać przekazując zestaw argumentów jak poniżej:

CallWebservice(<NAZWA INSTANCJI SERWISU>, <NAZWA METODY>, arg0, arg1, ... , argN)

gdzie:

NAZWA INSTANCJI SERWISU – nazwa identyfikująca zbiór parametrów systemowych stworzonych w bazie danych powiązanych z instancją serwisu (nazwa jest ustalana przy tworzeniu parametrów). Parametr obowiązkowy.

NAZWA METODY - nazwa identyfikująca zbiór parametrów systemowych stworzonych w bazie danych powiązanych z konkretną metodą instancji serwisu (nazwa jest ustalana przy tworzeniu parametrów). Parametr obowiązkowy.

arg0, arg1, ... , argN – opcjonalne (dodatkowe) argumenty podawane do funkcji podczas wykonywania reguły. Jako argumenty można podawać zmienne, pola oraz jawne wartości. Liczba podawanych argumentów jest dowolna. Można także nie podawać żadnych dodatkowych argumentów. Warto też zwrócić uwagę na numery argumentów (liczone od 0), ponieważ wartości tych dokumentów można wstawiać dynamicznie do parametrów systemowych zdefiniowanych w bazie.

3. Tworzenie wymaganego zestawu parametrów systemowych w bazie danych

Aby móc poprawnie wywołać żądanie do serwisu SOAP za pomocą metody CallWebservice obowiązkowo należy skonfigurować (dodać w bazie) odpowiedni zestaw customowych parametrów systemowych. Jest to konieczne, ponieważ AMODIT buduje żądanie do serwisu SOAP na podstawie danych zaciąganych z parametrów systemowych.

3.1. Zasady dodawania customowych parametrów:

1. Dodawanie nowego parametru oznacza dodanie nowego wiersza do tabeli **parameters**.
2. Nowych parametrów nie można póki co dodawać z poziomu aplikacji internetowej AMODIT, parametry należy dodawać ręcznie do bazy.
3. Dodając nowy wiersz do tabeli parameters należy wstawić wartości dla następujących kolumn:
 - **parName** – nazwa parametrów (o tym w dalszej części dokumentu)
 - **parValue** – wartość parametru (o tym w dalszej części dokumentu)
 - **parGroup** – nazwa grupy (nazwa kontenera widocznego na stronie z ustawieniami systemowymi, w którym umieszczone zostaną parametry zawierające nazwę tej samej grupy). Nazwa grupy powinna się zaczynać od 'Extension.' np. 'Extension.MyService123'. Dzięki prefiksowi Extension kontener

'MyService123' zawierający parametry będzie widoczny w Ustawieniach Systemowych w zakładce Extensions (Rozszerzenia).

- **parIndex** – numer ustalający kolejność parametrów widocznych wewnątrz kontenera (grupy parametrów), dowolne całkowite liczby nieujemne.
- **parType** – typ parametru. Dostępne są następujące typy:
 - 's' - tekstowy jednolinijkowy,
 - 't' - tekstowy wielolinijkowy (rozszerzalny edytor tekstu),
 - 'p' - hasło (niewidoczna zawartość),
 - 'b' – parametr typu checkbox.

3.2. Parametry niezbędne do konfiguracji nowej instancji serwisu

Na początek należy ustalić nazwę instancji serwisu. Przyjmijmy np. 'MyService123'.

Dla konfiguracji pojedynczej instancji serwisu należy dodać X parametrów. X wyliczamy wg poniższego wzoru:

$$X = 3 + (\text{Liczba_obsługiwanych_metod} * 2)$$

gdzie:

Liczba_obsługiwanych_metod – liczba metod (akcji) serwisu SOAP, które dla których docelowo będą wywoływane żądania z poziomu AMODIT.

Na podstawie powyższego wzoru widać, że zawsze dodajemy 3 „wspólne” parametry per instancja serwisu oraz dodatkowo po 2 „szczegółowe” parametry dla każdej metody z osobna.

Parametry wspólne:

Nazwa parametru (parName)	Wartość parametru (parValue)
external.ws.<NAZWA INSTANCJI SERWISU> np. <i>external.ws.MyService123</i>	URL prowadzący do lokalizacji pod którą jest uruchomiony docelowy serwis np. <i>https://mycompany.com/myservice.svc</i>
external.ws.<NAZWA INSTANCJI SERWISU>.Login np. <i>external.ws.MyService123.Login</i>	Nazwa użytkownika przysyłana w nagłówku uwierzytelniającym. Jeśli pusta wartość to żądanie jest przysyłane bez nagłówka. np. <i>JanKowalski123</i>
external.ws.<NAZWA INSTANCJI SERWISU>.Password np. <i>external.ws.MyService123.Password</i>	Hasło użytkownika przysyłane w nagłówku uwierzytelniającym. Jeśli pusta wartość to żądanie jest przysyłane bez nagłówka. np. <i>password123</i>

Parametry szczegółowe (2 parametry dla każdej z metod):

Nazwa parametru (parName)	Wartość parametru (parValue)
external.ws.<NAZWA INSTANCJI SERWISU>.SOAPAction np. <i>external.ws.MyService123.SOAPAction</i>	Nazwa wskazująca, która metoda z docelowego serwisu ma być wywołana np. <i>https://tempuri.org/IMyAPI/GetItem</i>
external.ws.<NAZWA INSTANCJI SERWISU>.Envelope np. <i>external.ws.MyService123.Login.Envelope</i>	Zawartość (paczka danych) wysyłania przy wywołaniu żądania. np. <pre><soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <soap:Body> <ItemQueryData> <ItemId>10</ItemId> </ItemQueryData> </soap:Body> </soap:Envelope></pre>

3.3 Dynamiczne wstawianie wartości do zawartości parametrów

AMODIT posiada opcję dynamicznego wstawiania wartości w treści pobieranie z parametrów systemowych podczas budowania żądania do serwisu SOAP. Z tego powodu tworzone parametry można nazywać także mianem szablonów. Operacja jest wykonywana automatycznie przy wywołaniu funkcji CallWebservice.

Przykład, wartość parametru Envelope:

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <ItemQueryData>
      <ItemId>{{0}}</ItemId>
      <ItemName>{{PoleA}}</ItemName>
      <ItemDate>{{ZmiennaB}}</ItemDate>
      <ItemType>Standard</ItemType>
    </ItemQueryData>
  </soap:Body>
</soap:Envelope>
```

W powyższym przykładzie mamy wartość parametru, która zawiera 3 placeholdery (wyróżniające się podwójnymi nawiasami falującymi, reprezentujące każdy z 3 możliwych typów placeholderów), pod które będą podstawione dynamicznie wartości przy budowaniu żądania do serwisu SOAP.

Placeholder {{0}} – typ I, wartość do podstawienia zostanie pobrana z argumentu *arg0* podawanego przy wywoływaniu funkcji. Analogicznie można tu odwołać się do dowolnego N-tego argumentu funkcji podając inną liczbę (*arg0*, *arg1*, ..., *argN*).

Placeholder {{PoleA}} – typ II, wartość do podstawienia pobierana jest z pola na sprawie (na której wykonywana jest reguła) o nazwie *PoleA*. W ramach podanego typu placeholdera można się odwołać do dowolnego pola na sprawie.

Placeholder {{ZmiennaB}} – typ III, wartość do podstawienia pobierana jest ze zmiennej (tutaj zmienna nosi nazwę *ZmiennaB*) zadeklarowanej w kontekście reguły gdzie wykonywana jest funkcja CallWebservice. W ramach placeholdera można podać nazwę dowolnej zmiennej zadeklarowanej w regule.

3.4 Dobre rady przy tworzeniu parametrów

- Przed konfiguracją parametrów w bazie AMODIT warto sprawdzić działanie serwisu np. w programie SOAPUI tworząc klienta serwisu na adresu url do kontraktu serwisu. Korzystając z wspomnianego programu można w łatwy sposób pozyskać np. wartości dla parametrów SOAPAction oraz Envelope.
- Niektóre serwisy SOAP wykorzystują swoje własne mechanizmy uwierzytelniania zamiast standardowych wbudowanych np. webserwis AMODIT - AMODWebAPI. W tym przypadku nie trzeba podawać wartości dla parametrów Login oraz Password.
- Jeśli XML będący docelową wartością parametrów Envelope jest obiektem złożonym (zawierającym dużo mniejszych obiektów) warto rozważyć opcję, gdzie elementarne obiekty mają swoje własne parametry (szablony) w bazie, skąd są pobierane i podstawiane są dla nich wartości, by następnie wstawić już do obiektu będącego wyżej w strukturze XML. W ten sposób docelowa struktura danych budowana jest od dołu do góry, a kod w regułach jest bardziej przejrzysty.
- Warto zwrócić uwagę na elementy (atrybuty XML) w wartości parametru Envelope, które są opcjonalne (wg opisu interfejsu serwisu) i zastanowić się czy bieżąca wartość dla parametru Envelope wspiera przypadek kiedy element występuje lub nie występuje w żądaniu.

4. Przydatne funkcje reguł:

- **XPath** – pobieranie ze struktury XML wskazanego elementu drzewa
- **GetSystemParameterValue** – pobieranie wartości parametru systemowego
- **StringFormat** – funkcja pozwala podmieniać wartości w ramach stringa (niezależnie od mechanizmu dynamicznego wstawiania działającego w ramach funkcji CallWebservice)

5. Dodatek: Konfiguracja parametrów pod serwis AMODWebAPI – uwagi

Do wywołania docelowego żądania na wybraną metodę w serwisie najpierw należy wykonać żądanie na metodę AuthenticateUser, która zwraca token (ważny przez 15 min).

Wspomniany token należy następnie przekazać wywołując żądanie na docelową metodę.