

eklient.piwet.pulawy.pl

Dokumentacja techniczna WEB SERVICE

Wersja 1.6 z dnia 27.06.2017

1. Wstęp

Niniejszy dokument specyfikuje interfejs API usług sieciowych (WebServices) umożliwiający nadawanie pakietów do systemu eklient należącego i administrowanego przez Państwowy Instytut Weterynaryjny – Państwowy Instytut Badawczy w Puławach. Komunikacja systemów informatycznych Klientów z systemem eklient odbywa się przy pomocy protokołu SOAP za pośrednictwem publicznego Internetu.

1.1. Strony komunikacji

W wymianie informacji uczestniczą dwie strony:

- Aplikacja eklient.piwet.pulawy.pl, pełniąca rolę serwera udostępniającego pewien zestaw funkcji, (nazywany w dalszej części Systemem). Dostępne usługi znajdują się pod adresem: <https://eklient.piwet.pulawy.pl/services>,
- System informatyczny Klienta uwierzytelniony za pomocą loginu i hasła, (nazywany w dalszej części Klientem).

1.2. Wymagania

System wymaga autoryzacji klienta dokonywanej poprzez dane uwierzytelniające login i hasło. W celu uzyskania danych autoryzacyjnych klient powinien zgłosić się do dostawcy usługi.

1.3. Komunikacja

Stroną aktywną podczas komunikacji jest Klient, który wysyła żądanie do Systemu i oczekuje na odpowiedź.

Usługi dostępne są pod adresem:
<https://eklient.piwet.pulawy.pl/services>

Szczegółowa specyfikacja WSDL dostępna jest pod adresem:
<https://eklient.piwet.pulawy.pl/services?wsdl>

2. Usługi udostępniane w ramach systemu

2.1. Pobranie zleceń

Metoda *getOrders* umożliwia pobranie zleceń zarejestrowanych w systemie i wysłanych do zautoryzowanego laboratorium. Metoda nie przyjmuje parametrów. W odpowiedzi system zwraca zestaw obiektów zawierających dane dostępnych zleceń.

Sygnatura metody:

ArrayOfZlecs getOrders()

Parametry wejściowe:

Brak

Zwracana wartość:

ArrayOfZlecs – tablica obiektów *Zlec* reprezentujących zlecenie lub obiekt błędu.

2.2. Pobranie zlecenia na podstawie identyfikatora protokołu

Metoda *getOrderByProtocolId* umożliwia pobranie zlecenia zarejestrowanego w systemie na podstawie numeru protokołu przekazanego jako parametr. W odpowiedzi zwracany jest obiekt zawierający dane zlecenia.

Sygnatura metody:

ArrayOfZlecs getOrderByProtocolId(*string* protocol_id)

Parametry wejściowe:

String protocol_id – Identyfikator protokołu

Zwracana wartość:

Zlec – obiekt *Zlec* reprezentujący zlecenie lub obiekt błędu

2.3. Potwierdzenie pobrania zleceń

Metoda *confirmOrdersUpdate* wysyła informację do systemu o prawidłowym pobraniu zleceń. Powinna być wywołana po każdym procesie zapisywania uprzednio pobranych zleceń. Brak potwierdzenia odbioru zapisanego zlecenia może skutkować dublowaniem zleceń w bazie klienta (niepotwierdzone zlecenie zostanie wysłane ponownie przy następnej próbie pobrania zleceń). Jako parametr metoda przyjmuje tablicę zawierającą numery id (numery id zleceń wg sytemu) poprawnie zapisanych zleceń. W odpowiedzi zwracana jest tablica identyfikatorów potwierdzonych zleceń.

Sygnatura metody:

ArrayOfIntegers confirmOrdersUpdate(*ArrayOfIntegers* tablica_id)

Parametry wejściowe:

ArrayOfIntegers – tablica identyfikatorów prawidłowo pobranych zleceń.
Uwaga: identyfikator zlecenia to numery id przesłany w obiekcie *Zlec* podczas pobierania zleceń!

Zwracana wartość:

ArrayOfIntegers – Tablica identyfikatorów potwierdzonych zleceń lub obiekt błędu.

2.4. Odrzucenie pobranego zlecenia

Metoda *rejectOrder* umożliwia odrzucenie przez laboratorium uprzednio pobranego zlecenia.

Sygnatura metody:

Boolean rejectOrder(*Integer* zlec_id, *String* message)

Parametry wejściowe:

Integer zlec_id – identyfikator odrzucanego zlecenia.

String message – przyczyna odrzucenia zlecenia.

Zwracana wartość:

Boolean – Wartość true jeśli odrzucenie zlecenia zostanie przyjęte w przeciwnym wypadku zwrócony zostanie obiekt błędu.

2.5. Pobranie zleceń oczekujących na wyniki

Metoda *getOrdersWaitingForResultFile* pobiera identyfikatory zleceń oczekujących na pliki raportów z wynikami.

Sygnatura metody:

ArrayOfIntegers getOrdersWaitingForResultFile()

Parametry wejściowe:

Brak

Zwracana wartość:

ArrayOfIntegers – tablica identyfikatorów zleceń oczekujących na wyniki

2.6. Przesłanie pliku raportu z badań

Metoda *uploadResultFile* pozwala wysłać plik wyniku dla danego zlecenia. Jeżeli istnieje potrzeba załączenia większej liczby plików metodę należy wywołać dla każdego pliku osobno. Funkcja akceptuje wyłącznie pliki pdf lub xades.

Sygnatura metody:

Boolean uploadResultFile(*Integer* zlec_id, *String* file, *String* filename)

Parametry wejściowe:

Integer zlec_id – identyfikator zlecenia

String file – plik zakodowany przy użyciu MIME base64

String filename – pełna nazwa pliku z rozszerzeniem(np. sprawozdanie.pdf)

Zwracana wartość:

Boolean – wartość true jeśli przesyłanie pliku zakończyło się sukcesem w przeciwnym wypadku zwrócony zostanie obiekt błędu

2.7. Pobranie korekt

Metoda *getComments* pozwala pobrać nowo wysłane korekty do zleceń. W odpowiedzi zwracana jest tablica obiektów korekt.

Sygnatura metody:

ArrayOfComments getComments()

Parametry wejściowe:

Brak

Zwracana wartość:

ArrayOfComments – tablica obiektów *Comments* lub obiekt błędu

2.8. Potwierdzenie pobranych korekt

Metoda *confirmCommentsUpdate* wysyła informację do systemu o prawidłowym pobraniu korekt. Powinna być wywołana po każdym procesie zapisywania uprzednio pobranych korekt. Brak potwierdzenia odbioru zapisanej korekty może skutkować dublowaniem korekt w bazie klienta (niepotwierdzona korekta zostanie wysłane ponownie przy następnej próbie pobrania korekt). Jako parametr metoda przyjmuje tablicę zawierającą numery id (numery id korekt wg systemu) poprawnie zapisanych korekt. W odpowiedzi zwracana jest tablica identyfikatorów potwierdzonych korekt.

Sygnatura metody:

ArrayOfIntegers confirmCommentsUpdate(*ArrayOfIntegers* confirm)

Parametry wejściowe:

ArrayOfIntegers – tablica identyfikatorów prawidłowo pobranych korekt. Uwaga: identyfikator korekty to numer id przesłany w obiekcie *Comment* podczas pobierania korekt!

Zwracana wartość:

ArrayOfIntegers – Tablica identyfikatorów potwierdzonych korekt lub obiekt błędu.

2.9. Dodanie korekty

Metoda *boolean* `addComments` pozwala wysłać nowe korekty do systemu.

Sygnatura metody:

boolean `addComments(ArrayOfComments comments)`

Parametry wejściowe:

ArrayOfComments – tablica obiektów `Comment`, zawierających dane korekt do dodania.

Zwracana wartość:

boolean – `TRUE` jeśli dodawanie korekt się powiodło, `FALSE` w przeciwnym wypadku lub obiekt błędu.

3. Struktury zdefiniowane w systemie

3.1. Błędy

Tablica opisująca błąd zwracany przez system.

Nazwa	Typ	Opis
faultcode	string	Rodzaj błędu
faultactor	string	Niewykorzystywane
faultstring	string	Kod błędu
detail	string	Opis błędu

Kody błędów:

Kod błędu	Opis błędu
1	Błędne dane autoryzacyjne.
2	Błąd argumentu

3.2. Unit

Opisuje firmę zarejestrowaną w systemie.

Nazwa	Typ	Opis
Id	integer	Identyfikator firmy
Name	string(255)	Nazwa firmy
Country	string(50)	Kraj
Nip	string(16)	Numer NIP firmy
AddressStreet	string(84)	Dane adresowe – ulica
AddressNumber	string(15)	Dane adresowe – numer budynku (lokalu)
AddressCity	string(255)	Dane adresowe – miejscowość
AddressPostcode	string(15)	Dane adresowe – kod pocztowy
AddressPost	string(255)	Dane adresowe – poczta
Email	string(255)	Adres email firmy
LogDd	dateTime	Data dodania
LogDe	dateTime	Data ostatniej zmiany danych

3.3. Comment

Opisuje komentarz zlecenia

Nazwa	Typ	Opis
Id	integer	Identyfikator korekty
ZlecId	integer	Identyfikator zlecenia
City	string(150)	Miejscowość utworzenia korekty
Message	string	Treść korekty
CreatedAt	dateTime	Data dodania
UpdatedAt	dateTime	Data ostatniej zmiany danych

3.3. DicData

Opisuje wartość słownikową.

Nazwa	Typ	Opis
Id	integer	identyfikator
Name	string(250)	wartość
Name2	string(250)	wartość dodatkowa

3.4. SampleXdata

Opisuje dodatkowe informacje o próbce (pola definiowane)**.

Nazwa	Typ	Opis
FieldId	integer	Identyfikator pola definiowanego
FieldName	string(150)	Nazwa pola definiowanego
Value	string	Wartość pola definiowanego dla typów nie słownikowych
DicValues	array	Tablica wartości słownikowych pola definiowanego[DicData]
Type	integer	Typ wyniku (1 - tekstowy, 2 - numeryczny, 3 - słownikowy, 4 - data, 5 - słownikowy wielokrotnego wyboru)
LogDd	dateTime	Data dodania informacji
LogDe	dateTime	Data ostatniej edycji informacji

3.5. ZlecXdata

Opisuje dodatkowe informacje o zleceniu.

Nazwa	Typ	Opis
Field	integer	Rodzaj dodatkowej informacji (1 – płatnik, 2 – adresat mailowy, 3 – adresat pocztowy, 4 -

		zleceniodawca)
Value	string	Dane firmy w momencie rejestracji zlecenia
*Unit	Unit	Dane firmy

3.6. Research

Określa kierunki i metody badawcze zlecenia.

Nazwa	Typ	Opis
Direction	integer	Identyfikator kierunku badania
Method	integer	Identyfikator metody badawczej

3.7. Sample

Opisuje próbkę.

Nazwa	Typ	Opis
Id	integer	Identyfikator rekordu próbki
Lp	integer	Numer próbki w ramach zlecenia
Ident	string(150)	Identyfikator próbki(podany przez zleceniodawcę)
Matrix	DicData	Rodzaj próbki
MatrixExt	string	Dodatkowe informacje nt rodzaju próbki
PobMiejsce	DicData	Miejsce pobrania
PobMiejsceExt	string(250)	Dodatkowe informacje nt miejsca pobrania
PobData	dateTime	Data pobrania
PobOsoba	string(200)	Osoba pobierająca
PobU	integer	Pobranie urzędowe (0 – nie, 1 – tak, 2 – nie dotyczy)
PobProc	string(250)	Procedura pobierania
PobPlan01	boolean	Pobranie zgodne z planem

PobPlan	string(250)	Plan pobierania
Kraj	string(3)	Kraj pochodzenia
Teryt	string(7)	Kod teryt miejsca pochodzenia
WlascicielTxt	string(250)	Właściciel
WlascicielExt	string(250)	Informacje dodatkowe nt właściciela
*Wlasciciel	Unit	Dane właściciela
ProducentTxt	string(250)	Producent
ProducentExt	string(250)	Informacje dodatkowe nt producenta
*Producent	Unit	Dane producenta
Latitude	decimal	szerokość geograficzna WGS 84
Longitude	decimal	długość geograficzna WGS 84
LogDd	dateTime	Data wprowadzenia informacji o rekordzie
LogDe	dateTime	Data ostatniej edycji informacji o rekordzie
Xdata	array	Tablica powiązanych informacji dodatkowych [SampleXdata]

3.8. Zlec

Opisuje zlecenie.

Nazwa	Typ	Opis
Id	integer	Identyfikator zlecenia
Kontynuacja	string(250)	Informacje o kontynuacji zlecenia
NazwaProj	string(250)	Nazwa projektu
BadTyp	DicData	Typ badania
BadCel	DicData	Cel badania
BadCelTxt	string(250)	Cel badania
SampleCnt	integer	Liczba próbek
PismaTow	string(250)	Pisma towarzyszące
NrZlecenia	string(250)	Numer protokołu pobrania próbek / nr zlecenia
Miasto	string(250)	Miejscowość wypełnienia protokołu

OpisProbki	string	Opis próbki
KierunekBadan	string	Wskazania dotyczące rodzaju badań
Uwagi	string	Uwagi
SporzName	string(30)	Osoba sporządzająca
SporzDate	dateTime	Data sporządzenia
LogDd	dateTime	Data wprowadzenia informacji
LogDe	dateTime	Data ostatniej aktualizacji informacji
Profil	integer	Profil zlecenia (1 – pasze, 2 - zoonozy)
*SporzOddzial	Unit	Dane sporządzającego
Samples	array	Tablica powiązanych próbek [Sample]
Xdata	array	Tablica powiązanych informacji dodatkowych [ZlecXdata]
Comments	array	Tablica powiązanych korekt [Comment]
Researches	array	Tablica kierunków i metod badania [Research]

* Dane firmy mogły ulec zmianie w czasie między rejestracją zlecenia a pobraniem go przez system klienta, obiekt Unit prezentuje aktualne dane firmy, podczas gdy w odpowiednim polu typu string prezentowany jest zrzut danych firmy z momentu rejestracji zlecenia.

** Słownik pól definiowanych oraz ich typów dostępny jest pod adresem http://www.piwet.pulawy.pl/celab-info/?cmd=pola_def.

4. Przykładowy klient w języku PHP (pseudokod)

```
<?php
// Inicjalizacja klienta soap i pobranie zleceń z webserwisów e-portal
$client = new nusoap_client('https://eklient.piwet.pulawy.pl/services?wsdl', true);
$client->soap_defencoding = 'UTF-8';
$client->decode_utf8 = false;
$client->setCredentials('login_xxxx', 'hasło_xxxx'); //uwierzytelnianie
$error = $client->getError();
```

```
if ($err) {  
    // ----- obsługa błędów -----  
}  
$proxy = $client->getProxy();  
  
$result = $proxy->getOrders(); // pobranie zleceń z serwisu e-portal  
  
// --- zapis pobranych zleceń w bazie klienta, ewentualna obsługa błędów ---  
  
$proxy->confirmOrdersUpdate($przyjete); // potwierdzenie odebranych zleceń
```

5. Zmiany w dokumentacji

Wersja 1.1

- Zmiana struktury Sample, dodane zostały pola Latitude i Longitude

Wersja 1.2

- Dodanie struktury Comment
- Dodanie struktury ArrayOfComments
- Zmiana struktury Zlec, dodane zostało pole Comments
- Dodanie metody getComments, pobierającej korekty zleceń
- Dodanie metody confirmCommentsUpdate, potwierdzającej pobranie korekt
- Dodanie metody addComments, dodającej nowe korekty do zleceń
- Zmiana struktury Unit, rozdzielenie pola Address na dwa pola AddressStreet i AddressNumber

Wersja 1.3

- Zmiana struktury Sample, zmiana typu pola PobU

Wersja 1.3.1

- Dodanie Zleceniodawcy do możliwych wartości pola Field w strukturze ZlecXdata

Wersja 1.4

- Usunięcie ze struktury Sample pola Sprzedawca
- Zmiana struktury Unit, dodanie pola Country

Wersja 1.5

- Dodanie pola 'kontynuacja' w strukturze zlecenia

Wersja 1.6

- Dodanie metody rejectOrder