

**„Rezerwacje Online i Obsługa Płatności”
(ROOP)
Dokumentacja techniczna
„Interfejs Serwisu Klienta”**

Wersja 1.9.3000 (ostatnia modyfikacja : 2011-09-05 MP)

Spis treści

1. Opis systemu ROOP :.....	2
Cel systemu.....	2
2. Obsługa systemu.....	3
Wywołanie formularza rezerwacji (easyReservation).....	3
Obsługiwane parametry wymagane :.....	3
Obsługiwane parametry opcjonalne :.....	3
Panel agenta.....	5
Logowanie.....	5
Lista rezerwacji.....	6
3. „Interfejs Serwisu Klienta” API.....	7
Uproszczony diagram przepływu danych w systemie ROOP.....	7
PHP API (classRoop.php).....	8
Pola klasy classRoop.....	8
Metody klasy classRoop.....	8
Funkcja test() : boolean.....	8
Funkcja login() : boolean.....	8
Funkcja logout() : boolean.....	9
Funkcja checkAvailability (TOfferId) : array().....	9
Funkcja getOfferDetails (TOfferId) : TOfferDetails.....	10
Funkcja createQuery(TQuery) : TTransaction.....	11
Funkcja createReservation(TReservation) : TTransaction.....	11
Funkcja acceptTransaction (transactionId) : TTransaction.....	12
Funkcja getTransaction (transactionId) : TTransaction.....	12
Funkcja statusTransaction (transactionId) : transactionId.....	12
Funkcja statusReservation (transactionId) : TTransaction.....	13
Funkcja getTransactionList (filters) : TTransactions.....	13
Nagłówek klasy classRoop.....	14
Złożone typy danych.....	15
TPerson.....	15
TParticipants.....	16
TAdditionalService.....	17
TAdditionalServices.....	18
TOfferId.....	18
TOfferDetails.....	19
TQuery.....	21
TReservation.....	21
TTransaction.....	22
TPriceParticipant.....	22
TPayment.....	23
TPayments.....	23
Dodatkowe informacje techniczne dla API.....	24
Ogólny opis standardów:.....	24
Wymagania techniczne:.....	24
Kody błędów.....	24

1. Opis systemu ROOP :

Cel systemu

Celem systemu jest udostępnienie agencjom turystycznym możliwości łatwej realizacji rezerwacji online z autorskich stron systemów internetowych, wyszukiwarek lub systemów CRM. System pozwala na całkowite przekazanie obsługi rezerwacji na stronę ROOP włącznie z obsługą płatności online i obsługą klienta dla rezerwacji, które dotychczas wymagały pracy „ręcznej”. Agent będzie mógł sam zdecydować i skonfigurować w panelu administracyjnym, które rezerwacje mają być obsługiwane ręcznie (informacja na email) a które będzie obsługiwał automat wraz z poprowadzeniem klienta przez całą ścieżkę rezerwacji i płatności.

Mechanizmem, który zostanie udostępniony agentowi będzie API. API udostępni agentowi wszystkie niezbędne narzędzia do przeprowadzenia rezerwacji online dla dowolnej znalezionej imprezy turystycznej w ramach systemów MDS. Udostępnione są także narzędzia pozwalające na monitorowanie wszystkich otwartych i zakończonych transakcji.

2. Obsługa systemu

Wywołanie formularza rezerwacji (easyReservation)

Najprostszą metodą użycia systemu rezerwacji ROOP jest użycie szybkiego wywołania formularza rezerwacji. Ze strony agenta niezbędne jest jedynie posiadanie :

- aktywnej wykupionej o opłaconej usługi MDSWS,
- loginu i hasła do MDSWS,
- **posiadanie podpisanej umowy z systemem płatności dotPay** (w przyszłości także inne),
- serwisu internetowego działającego w oparciu o technologię MDS lub MerlinX IBE.

Obsługiwane parametry wymagane :

- offerIdType : należy przekazać stałą wartość MDS,
- offerIdHash : należy przekazać unikalny identyfikator oferty z MDS,
- agentId : numer agenta z systemu MerlinX (9-cio cyfrowy),
- adults : ilość osób dorosłych,
- children : ilość dzieci,
- infants : ilość dzieci poniżej 2 roku życia,

Obsługiwane parametry opcjonalne :

- Lista parametrów, za pośrednictwem których można do formularza przelać dane :

płatnika :

platnik_plec (char(1) = m | f)
platnik_imie (char*)
platnik_nazwisko (char*)
platnik_ulica_nr_lok (char*)
platnik_kod_pocztowy (char*)
platnik_data_urodzenia (char format YYYY-MM-DD)
platnik_miejscowosc (char*)
platnik_kraj (char*)
platnik_region (char*)
platnik_telefon_1 (char*)
platnik_email (char* - email)

uczestników :

uczestnik_0_plec (char(1) = m | f)
uczestnik_0_imie (char*)
uczestnik_0_nazwisko (char*)
uczestnik_0_data_urodzenia (char* format YYYY-MM-DD)
uczestnik_0_paszport (char*)
uczestnik_0_ulica_nr_lok (char*)
uczestnik_0_kod_pocztowy (char*)
uczestnik_0_miejscowosc (char*)
uczestnik_0_region (char*)

uczestnik_0_telefon_1 (char*)
uczestnik_0_email (char* - email)

uczestnik_1_plec (char(1) = m | f)
uczestnik_1_imie (char*)
uczestnik_1_nazwisko (char*)
uczestnik_1_data_urodzenia (char* format YYYY-MM-DD)
uczestnik_1_paszport (char*)
uczestnik_1_ulica_nr_lok (char*)
uczestnik_1_kod_pocztowy (char*)
uczestnik_1_miejscowosc (char*)
uczestnik_1_region (char*)
uczestnik_1_telefon_1 (char*)
uczestnik_1_email (char* - email)
(...)
uczestnik_n_*

- template : wybór szablonu graficznego formularza (wymaga kontaktu z www.ksimedia.pl),
- affiliateId : identyfikator afiliatu, z którego wywołano formularz (**w przygotowaniu**).

Przykład wywołania:

<http://roop.twojeplatnosci.com.pl/reservation.html&offerIdType=MDS&offerIdHash=8b14737cfaebd9c80232a4e8d6598f4eb6b384f2e34141b1bf05967faaacff&agentNo=193349332>

Z danymi płatnika i uczestników:

http://roop.twojeplatnosci.com.pl/index.php?module=roop/dialog/reservation&offerIdType=MDS&offerIdHash=f28d73ac74d19ceef95c4eaa2284cfa085c7da0349b1cb17d83700abd6b10b22&adults=2&children=0&infants=0&agentId=809442001¶ms_adults=2¶ms_children=1¶ms_infants=0&platnik_plec=m&platnik_imie=Adam&platnik_nazwisko=Nowak&platnik_ulica_nr_lok=Nowakowska+10&platnik_kod_pocztowy=12-345&platnik_data_urodzenia=1948-06-24&platnik_miejscowosc=Nowakowo&platnik_kraj=Polska&platnik_region=slaskie&platnik_telefon_1=123456789&platnik_email=nowak@gmail.com&uczestnik_0_plec=m&uczestnik_0_imie=Jan&uczestnik_0_nazwisko=Nowak&uczestnik_0_data_urodzenia=1958-08-24&uczestnik_0_ulica_nr_lok=Nowakowska+19&uczestnik_0_kod_pocztowy=12-345&uczestnik_0_miejscowosc=Chorzow&uczestnik_0_telefon_1=123456789&uczestnik_0_email=email@nowak.com&uczestnik_1_plec=f&uczestnik_1_imie=Anna&uczestnik_1_nazwisko=Nowak&uczestnik_1_data_urodzenia=1961-03-30&uczestnik_1_ulica_nr_lok=Nowakowska+15&uczestnik_1_kod_pocztowy=12-400&uczestnik_1_miejscowosc=Chorzow&uczestnik_1_telefon_1=987654321&uczestnik_1_email=email@ona.pl

Uwagi:

Zalecane jest wywoływanie formularza i przekazywanie parametrów metodą POST.
W przypadku przekazywania parametrów metodą GET **ważne jest aby za nazwą reservation.html użyć znak rozdzielający '&' zamiast '?'** (tak jak w przykładzie).

Panel agenta

Panel agenta dostępny jest pod adresem : <http://twojeplatnosci.com.pl/panel.html>.

Logowanie

Włączenie Panelu agenta wymaga zalogowania się. Login i hasło są takie same jak login i hasło do systemu MDSWS.



The image shows a login form with a blue header bar containing the text "LOGOWANIE". Below the header, there are two input fields: the first is labeled "Login" and the second is labeled "Hasło". Below these fields is a blue button with the text "Zaloguj".

Możliwe jest także pominięcie okna logowania (tajemnica MX, KSI)

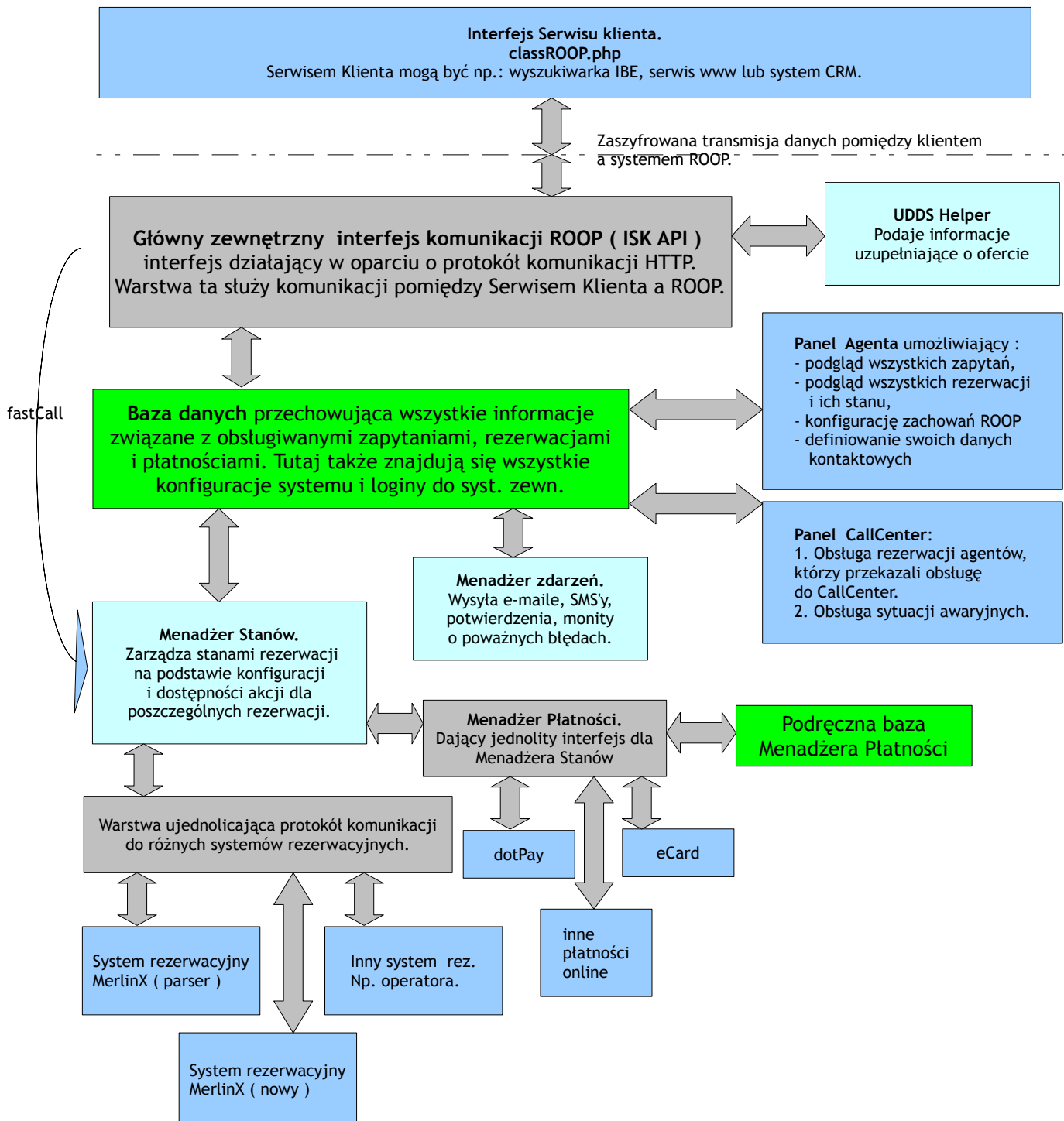
Lista rezerwacji

Lista rezerwacji Dodaj nowa rezerwacje Konfiguracja Profil Wyloguj								
Lista transakcji								
Typ	---	Wybór typu	Status	---	Wybór statusu	Data od	Data do	<input type="button" value="Filtruj"/>
Znaleziono 9 otwartych płatności								
Nazwisko / imię ▲▼	Obsługa ▲▼	Status ▲▼	Wyjazd ▲▼	Powrót ▲▼	Opcja do ▲▼	Organizator ▲▼	Nr rez. ▲▼	Wartość ▲▼
Nowak / Marcin	auto	do_akceptacji	28.09.2011	05.10.2011	2010-11-28 16:38:00	AQUA		2 089,00 PLN
Szczański / Maciej	auto	zakonczone	28.09.2011	05.10.2011	2010-11-28 16:55:00	AQUA	110143	2 284,00 PLN
Koń / Adam	auto	do_akceptacji	12.09.2011	19.09.2011	2010-11-28 16:38:00	AQUA		1 532,00 PLN
Kielbas / Rafał	auto	do_akceptacji	07.09.2011	14.09.2011	2010-11-28 16:31:00	AQUA		2 733,00 PLN
Kielbas / Rafał	auto	do_akceptacji	07.09.2011	14.09.2011	2010-11-28 16:37:00	AQUA		2 733,00 PLN
Kielbas / Rafał	auto	do_akceptacji	07.09.2011	14.09.2011	2010-11-28 16:44:00	AQUA		2 733,00 PLN
nowak / Adam	auto	zakonczone	24.08.2011	31.08.2011	2010-11-28 16:38:00	AQUA	110142	2 549,00 PLN
Przybylski / Franciszek	auto	do_akceptacji	27.01.2011	30.01.2011	2010-11-27 16:38:00	ORKA		4 647,00 PLN
Przybylski / Franciszek	auto	zakonczone	27.01.2011	30.01.2011	2010-11-27 17:55:00	ORKA	64400	5 867,00 PLN
UCZESTNICZY (3)			PŁATNOŚĆ			CENA ZAWIERA		
Przybylski / Franciszek		01.11.1980	online	2010-11-26 17:54:40	1 466,75 PLN	Kategoria: 0		
Przybylska / Anna		01.11.1980			1 466,75 PLN	Wylot z / do: /		
Przybylski / Guccio		01.11.2005				Rodzaj pokoju:		
						Wyżywienie:		
USŁUGI DODATKOWE			INFORMACJE DODATKOWE			KIERUNEK		
			Data rezerwacji: 26.11.2010					
			Utworzono: 09.11.2010 22:00:00					
			Zmodyfikowano: 26.11.2010 18:01:40					
<input type="button" value="Zobacz rezerwację"/>			<input type="button" value="Zakończ"/>					

3. „Interfejs Serwisu Klienta” API

Dla użytkowników zaawansowanych przygotowano API pozwalające na samodzielne zbudowanie i oprogramowanie formularza rezerwacji.

Uproszczony diagram przepływu danych w systemie ROOP



PHP API (classRoop.php)

Pola klasy classRoop

- myLogin : string,
- myPassword : string,
- lastErrorId : int,
- lastErrorDescription : string.

Metody klasy classRoop

Funkcja test() : boolean

Opis :

Funkcja do testowania połączenia z serwerem ROOP. Nie wymaga loginu i hasła.

Wejście :

- nie wymaga

Wyjście :

Komunikat o wersji API działającej po stronie serwera oraz „echo” wersji klienta, który wysłał komendę test();
przykład odpowiedzi:

```
KSI Media ( www.ksimedia.pl ). ROOP API SERVER Version : 1.0.0.1000. Your client version is : 1.0.0.1021.
```

Funkcja login() : boolean

Opis :

Funkcja logowania do systemu ROOP. Wykonywanie jakichkolwiek innych operacji w ramach systemu wymaga zalogowania. Sesja pozostaje aktywna do zakończenia działania skryptu po stronie klientna.

Wejście :

- this->myLogin,
- this->myPassword.

Wyjście :

W zależności od tego czy logowanie do systemu się powiodło funkcja zwraca wartość true lub false. W przypadku zwrotu false, numer błędu i opis znajdują się w polach klasy :
this->lastErrorId oraz this->lastErrorDescription.

Uwagi :

- Login i hasło do systemu ROOP pozyskiwane są w drodze podpisania odpowiedniej umowy z MerlinX na korzystanie z usługi „ ” .
- Zerwanie komunikacji lub zakończenie po stronie serwera nie powoduje wywołania specjalnej akcji po stronie interfejsu. Sytuacja taka będzie sygnalizowana błędem #001 lub #002.

Funkcja logout() : boolean

Opis :

Funkcja pozwala na zakończenie *sesji* po stronie serwera ROOP.

Wyjście :

W zależności od tego czy wylogowanie z systemu się powiodło funkcja zwraca wartość true lub false.

W przypadku zwrotu false, numer błędu i opis znajdują się w polach klasy :
this->lastErrorId oraz this->lastErrorDescription.

Funkcja checkAvailability (TOfferId) : array()

Opis :

Funkcja pozwala na sprawdzenie dostępności oferty i możliwości wykonania na tej ofercie dalszych operacji. Funkcja jest mało obciążająca dla systemu w związku z czym zalecane jest jej użycie do sprawdzanie dostępności na listach terminów.

Wejście :

- TOfferId
struktura danych jednoznacznie definiująca ofertę, której dostępność chcemy sprawdzić.

Wyjście:

result [availability] :

Możliwe odpowiedzi funkcji typu string :

'x' - impreza nie jest dostępna,

'r' - możliwe jest dalsze prowadzenie rezerwacji online,

result [priceAdult] : w groszach (opcjonalne)

result [priceTotal] : w groszach (opcjonalne)

result [prices] : lista z cenami uczestników

w przypadku zwrotu false, numer błędu i opis znajdują się w polach klasy :
this->lastErrorId oraz this->lastErrorDescription.

Funkcja `getOfferDetails (TOfferId)` : `TOfferDetails`

Opis:

Funkcja zwraca kompletną informację dotyczącą imprezy, którą chcemy zarezerwować. Funkcji należy używać do wyświetlania informacji o ofercie na formularzu zapytania lub rezerwacji.

Wejście:

- `TOfferId`
struktura danych jednoznacznie definiująca ofertę, której dostępność chcemy sprawdzić,
Pola opcjonalne w niektórych przypadkach pozwalają zaprezentować dokładniejszą cenę całkowitą jeszcze na etapie generowania ogólnych informacji o ofercie.

Wyjście:

- `TOfferDetails`
Struktura zawierająca szczegóły dotyczące oferty.

w przypadku zwrotu `false`, numer błędu i opis znajdują się w polach klasy :
`this->lastErrorId` oraz `this->lastErrorDescription`.

Funkcja createQuery(TQuery) : TTransaction

Opis:

Funkcja tworzy nowe zapytanie klienta w systemie ROOP. W zależności od konfiguracji „Panelu Agenta” zapytanie zostanie obsłużone przez CallCenter lub zostanie wysłane na wskazany w konfiguracji agenta adres e-mail lub SMS.

Wejście:

- TQuery
struktura danych zawierająca niezbędne informacje do zbudowania zapytania w systemie ROOP.

Wyjście:

- TTransaction
Struktura zawierająca szczegóły dotyczące założonej transakcji oferty.

w przypadku zwrotu `false`, numer błędu i opis znajdują się w polach klasy :
`this->lastErrorId` oraz `this->lastErrorDescription`.

Funkcja createReservation(TReservation) : TTransaction

Opis:

Funkcja tworzy nową rezerwację w systemie ROOP. W zależności od konfiguracji „Panelu Agenta” rezerwacja zostanie obsłużona przez CallCenter zostanie obsłużona przez automat.

Wykonanie funkcji jest jednoznaczne z założeniem opcji lub rozpoczęciem transakcji w systemie organizatora. Zdarzenie, które nastąpi jest uzależnione od konfiguracji w „Panelu Agenta” oraz możliwości wykonania rezerwacji lub opcji na danej ofercie.

Wejście:

- TReservation
struktura danych zawierająca niezbędne informacje do zbudowania rezerwacji w systemie ROOP.

Wyjście:

- TTransaction
Struktura zawierająca szczegóły dotyczące założonej transakcji oferty.

w przypadku zwrotu `false`, numer błędu i opis znajdują się w polach klasy :
`this->lastErrorId` oraz `this->lastErrorDescription`.

Uwagi :

Wykonanie funkcji powoduje fizyczne założenie opcji lub rozpoczęcie transakcji sprzedaży.
Upewnij się, że jej wywołanie nie jest przypadkowe.

Funkcja acceptTransaction (transactionId) : TTransaction

Opis:

Funkcja wysyła do ROOP żądanie zatwierdzenia transakcji. Oznacza to, że klient już zapoznał się z ceną końcową i chce przejść do procesu płatności.

Wejście:

- transactionId : string
unikalny identyfikator transakcji. Nadawany jest podczas createQuery() lub createReservation().

Wyjście:

- TTransaction
Struktura zawierająca szczegóły dotyczące założonej transakcji oferty.

w przypadku zwrotu **false**, numer błędu i opis znajdują się w polach klasy :
this->lastErrorId oraz this->lastErrorDescription.

Funkcja getTransaction (transactionId) : TTransaction

Opis:

Funkcja zwraca komplet informacji związanych z transakcją prowadzoną przez system ROOP.

Wejście:

- transactionId : string
unikalny identyfikator transakcji. Nadawany jest podczas createQuery() lub createReservation().

Wyjście:

- TTransaction
Struktura zawierająca szczegóły dotyczące założonej transakcji oferty.

w przypadku zwrotu **false**, numer błędu i opis znajdują się w polach klasy :
this->lastErrorId oraz this->lastErrorDescription.

Funkcja statusTransaction (transactionId) : transactionId

Opis:

Funkcja uruchamia przetwarzanie transakcji po stronie serwera i zwraca komplet informacji związanych z transakcją prowadzoną przez system ROOP.

Wejście:

- transactionId : string
unikalny identyfikator transakcji. Nadawany jest podczas createQuery() lub createReservation().
Dopuszczalna wartość 'all' - przetwarzanie dla wszystkich transakcji w systemie. Funkcjonalność ta nie jest dostępna dla klienta ROOP.

Wyjście:

- transactionId : string
Struktura jako potwierdzenie poprawności zwraca przekazane transactionId.

w przypadku zwrotu **false**, numer błędu i opis znajdują się w polach klasy :
this->lastErrorId oraz this->lastErrorDescription.

Funkcja statusReservation (transactionId) : TTransaction

Opis:

Funkcja uruchamia przetwarzanie transakcji po stronie serwera i zwraca komplet informacji związanych z transakcją prowadzoną przez system ROOP - dane uzupełnione o bieżące informacje z systemu tour operatora.

[funkcja w przygotowaniu]

Wejście:

- transactionId : string
unikalny identyfikator transakcji. Nadawany jest podczas createQuery() lub createReservation().

Wyjście:

- transactionId : string
Struktura jako potwierdzenie poprawności zwraca przekazane transactionId.

w przypadku zwrotu **false**, numer błędu i opis znajdują się w polach klasy :
this->lastErrorId oraz this->lastErrorDescription.

Funkcja getTransactionList (filters) : TTransactions

Opis:

Funkcja zwraca listę zamówień złożonych do systemu ROOP.

[funkcja w przygotowaniu]

Wejście:

- filters : Tfilters
określa parametry, wg których zostaną wybrane zamówienia/transakcje z systemu ROOP.

Wyjście:

- TTransactions : array
lista transakcji pobranych z roop.

w przypadku zwrotu **false**, numer błędu i opis znajdują się w polach klasy :
this->lastErrorId oraz this->lastErrorDescription.

Nagłówek klasy classRoop

```
class classRoop {  
    public $myLogin = '' ;  
    public $myPassword = '' ;  
    public $lastErrorId = 0 ;  
    public $lastErrorDescription = '' ;  
  
    function login() ;  
    function logout() ;  
    function checkAvailibility ( $offerId ) ;  
    function getOfferDetails ( $offerId ) ;  
    function createQuery ( $query ) ;  
    function createReservation ( $reservation ) ;  
    function getTransaction ( $transactionId ) ;  
    function statusTransaction ( $transactionId ) ;  
    function getDictionary ( $dictionaryName ) ; ( w budowie )  
}
```

Przykład uruchomienia klasy :

```
<?php  
  
    require_once ( 'classRoop.php' ) ;  
  
    // Logowanie  
    $roop = new classRoop();  
    $roop->myLogin = 'myAgentLogin' ;  
    $roop->myPogin = 'myAgentPassword' ;  
    if ( $roop->login() == false ) {  
        echo 'Błąd podczas logowania : ' ;  
        echo $roop->lastErrorDescription ;  
        exit ;  
    } else {  
        echo 'Witam, jesteś zalogowany.' ;  
    }  
  
    // Pobieram informacje o transakcji  
    print_r ( $roop->getTransaction ( '7AYHBD56W99EMAVD23TS9876YUSHG3NJ' ) ) ;  
  
?>
```

Złożone typy danych

Wszystkie złożone typy danych są realizowane za pomocą tablic PHP.

TPerson

Opis pól:

- [gender] char(1) : dopuszczalne wartości 'm' | 'f'.
pole określa płeć osoby m = male (mężczyzna), f = female (kobieta),
- [name] char(45) : imię osoby,
- [surName] char(45) : nazwisko osoby,
- [birthDate] char(10) : data urodzenia w formacie YYYY-MM-DD,
- [passportNo] char(45) : numer paszportu,
- [documentNo] char(45) : numer dokumentu tożsamości,
- [peselNo] char(20) : numer PESEL,
- [phoneA] char(30) : numer telefonu, przykład : +48.224562343,
- [phoneB] char(30) : dodatkowy numer telefonu, przykład : +48.224562343,
- [emailA] char(45) : adres email,
- [emailB] char(45) : dodatkowy adres email,
- [address] char(80) : ulica i numer lokalu, przykład : „Powstańców 20/32”,
- [city] char(80) : miejscowość,
- [postCode] char(45) : kod pocztowy,
- [country] char(45) : kraj,
- [region] char(45) : region / województwo (wszystkie polskie lub 'inne'),
- [children] pole opcjonalne ROOP zwraca w nim czy osoba w danej ofercie jest dzieckiem,
wartości : 0 = dorosły, 1 = dziecko, 2 = niemowlę.

Przykład:

```
Array (  
    [gender] => m  
    [name] => Tadeusz  
    [surName] => Kowalski  
    [birthDate] => 1940-02-14  
    [passportNo] => PL1111111111111111  
    [documentNo] => APP 1234567  
    [peselNo] => 37043022221  
    [phoneA] => +48.277777777  
    [phoneB] => +48609111111  
    [emailA] => email@email.pl  
    [emailB] => emaildwa@email.pl  
    [address] => Nowaka 3/12  
    [city] => Koszalin  
    [postCode] => 12-345  
    [country] => Poland  
)
```

TParticipants

Typ opisujący listę uczestników imprezy. Jest tablicą, której każdy z elementów jest typem TPerson.

Przykład:

```
Array
(
    [0] => Array
        (
            [gender] => m
            [name] => Tadeusz
            [surName] => Kowalski
            [birthDate] => 1930-02-14
            [passportNo] => PL1111111111111111
            [documentNo] => APP 1234567
            [peselNo] => 1111111111
            [phoneA] => +48.111111111
            [phoneB] => +48111111111
            [emailA] => email@emial.pl
            [emailB] => emaildwa@email.pl
            [address] => Nowaka 3/12
            [city] => Koszalin
            [postCode] => 12-345
            [country] => Poland
        )
    [1] => Array
        (
            [gender] => f
            [name] => Janina
            [surName] => Kowalska
            [birthDate] => 1952-07-11
            [passportNo] => PL1111111111111111
            [documentNo] => APP 11111111
            [peselNo] => 1111111111
            [phoneA] => +48.111111111
            [phoneB] => +48111111111
            [emailA] => email@emial.pl
            [emailB] => emaildwa@email.pl
            [address] => Nowaka 3/12
            [city] => Koszalin
            [postCode] => 12-345
            [country] => Poland
        )
)
```


TAdditionalService

Typ opisujący strukturę serwisu / dodatków do rezerwacji. Pojedynczy serwis może być przypisany do rezerwacji lub do osoby w ramach rezerwacji.

Struktura jest używana zarówno jako struktura informacyjna dla klienta końcowego jak również struktura służąca do zamawiania usług.

Opis pól:

- [code] char(32) : kod dla danego serwisu (nadawany przez ROOP),
- [type] char(16) : typ usługi dodatkowej możliwe wartości : wish | service,
- [name] char(128) : potoczna nazwa usługi dodatkowej,
- [price] int : cena za usługę dodatkową w groszach,
- [participant_id] string : jeśli usługa dodatkowa jest przypisana do uczestnika imprezy to w tym polu jest umieszczony identyfikator elementu (key) tablicy Tparticipants jeśli dotyczy całej transakcji pole to jest stringiem pustym lub null.
- [quantity] int : ilość zamówionych usług,
pole jest opcjonalne i jego występowanie zależy od rodzaju zamówionej usługi.
- [dateFrom] char(10) : data od kiedy wykupiona usługa obowiązuje, format daty : YYYY-MM-DD,
pole jest opcjonalne i jego występowanie zależy od rodzaju zamówionej usługi.
- [dateTo] char(10) : data do kiedy wykupiona usługa obowiązuje, format daty : YYYY-MM-DD,
pole jest opcjonalne i jego występowanie zależy od rodzaju zamówionej usługi.

Przykład :

Gdy otrzymujemy jako info o możliwych do zamówienia usługach w imprezie:

```
Array (  
  [code] => u01  
  [type] => wish  
  [name] => ubezpieczenie od nagłej śmierci pod prysznicem  
  [price] => 14300  
)
```

Gdy otrzymujemy jako info o zamówionych w rezerwacji dodatkach:

```
Array (  
  [code] => u01  
  [type] => wish  
  [name] => ubezpieczenie od nagłej śmierci pod prysznicem  
  [price] => 14300  
  [participant_id] => 0  
  [quantity] => 1  
  [dateFrom] => 2010-11-12  
  [dateTo] => 2010-11-25  
)
```

Gdy wysyłamy / zamawiamy - createReservation():

```
Array (  
  [code] => u01  
  [type] => service  
  [name] => ubezpieczenie od nagłej śmierci pod prysznicem  
  [participant_id] =>  
  [quantity] =>  
  [dateFrom] =>  
  [dateTo] =>  
)
```

TAdditionalServices

Typ opisujący listę uczestników imprezy. Jest tablicą, której każdy z elementów jest typem TAdditionalService.

Przykład :

raczej zbędny.

TOfferId

Struktura, której zawartość jednoznacznie określa ofertę w systemie rezerwacyjnym, wyszukiwania i pośrednio w systemie organizatora.

Opis pól:

- [type] char(32) : rodzaj kodu określającego imprezę
możliwe wartości typu string :
 - 'MDS'
 - 'XML'

W przypadku typu MDS dodatkowe pola to :

- [hash] : string pozyskany z MDS,
- [adults] : int ilość osób dorosłych,
- [children] : int ilość dzieci,
- [infants] : int ilość dzieci poniżej 2 roku życia,

W przypadku typu XML dodatkowe pola to (obecnie nie dotyczy) :

- [.....] : string

Przykład dla MDS:

```
Array (  
    [type] => MDS  
    [sHash] => 7cc88bd0dce5e9202144c0b16239650f  
    [uHash] => 935005810b0e3ff2958065e7f8411bdc  
)
```

Przykład dla XML (do uzupełnienia) :

```
Array (  
    [type] => XML  
    [tourOp] => ALFA  
    [catalog] => LAST  
    [hotelCode] => HPRO92S  
    ( ... )  
)
```

TOfferDetails

Struktura zawiera informacje uzupełniające na temat oferty.

Opis pól :

- [offerName] string : Tytuł oferty / nazwa hotelu,
- [category] int : kategoria obiektu (pomnożona przez 10)
- [country] string : nazwa kraju (lub objazdowe),
- [region] string : nazwa regionu (lub objazdowe),
- [city] string : nazwa miejscowości (lub objazdowe),
- [departureDate] string : data wyjazdu / wylotu (YYYY-MM-DD),
- [returnDate] string : data powrotu (YYYY-MM-DD),
- [transport] string : samolot | autokar | pociag | własny,
- [departureFrom] string : wyjazd / wylot z miejscowości (opcja tylko dla samolot, autokar),
- [departureTo] string : przelot / przewóz do miejscowości,
- [dapartureTime] array() : [depTime], [arrTime] (format GGSS),
- [arrivalDate] string : data przylotu (powrotu),
- [arrivalTime] array() : [depTime], [arrTime] (format GGSS),
- [feeding] string : rodzaj żywienia,
- [roomType] string : rodzaj pokoju,
- [length] string : długość pobytu,
- [minPerson] string : minimalna ilość osób,
- [maxPerson] string : maksymalna ilość osób,
- [minAdult] string : minimalna ilość osób dorosłych,
- [maxAdult] string : maksymalna ilość osób dorosłych,
- [priceAdult] int : cena za jedną osobę dorosłą w groszach,
- [additionalServices] TAdditionalServices : dostępne usługi dodatkowe,
- [availability] string : dostępność oferty zgodnie z funkcją checkAvailability,
- [tourOperator] : string : kod organizatora,
- [tourOperatorConditionsURL] : string : url do warunków uczestnictwa organizatora,

Przykład:

```
Array (  
  [offerName] => Golden Beach  
  [category] => 35  
  [country] => Egipt  
  [region] => Hurghada  
  [city] => Hurghada  
  [departureDate] => 2011-08-12  
  [returnDate] => 2011-08-26  
  [transport] => samolot  
  [departureFrom] => Katowice  
  [departureTo] => HRG  
  [dapartureTime] => 12:46  
  [arrivalDate] =>  
  [arrivalTime] =>
```

```
[feeding] => Śniadania i obiadowe kolacje
[roomType] => DBL+2 dostawka
[length] => 14
[minPerson] => 2
[maxPerson] => 4
[minAdult] => 2
[maxAdult] => 3
[priceAdult] => 245000
[additionalServices] => Array (
    ***
)
[availability] => o
[tourOperator] => NPL
)
```

TQuery

Struktura zawiera informacje, które są niezbędne do założenia zapytania w systemie ROOP.

Opis pól :

- [offerId] TofferId
- [payer] TPerson
- [participants] Tparticipants : (opcjonalne)
- [additionalServices] TadditionalServices : (opcjonalne)

TReservation

Struktura zawiera informacje, które są niezbędne do założenia rezerwacji w systemie ROOP.

Opis pól :

- [offerId] TofferId
- [payer] TPerson
- [participants] Tparticipants
- [additionalServices] TadditionalServices

TTransaction

Struktura zawierająca komplet informacji związanych z prowadzoną przez system ROOP transakcją. Transakcją mogą być : zapytanie, opcja, rezerwacja.

Opis pól :

- [transactionId] string : identyfikator transakcji w systemie ROOP (niezmienny),
- [transactionNumber] string : „przyjazny” numer transakcji (unikalny w ramach agenta),
- [type] string : rodzaj możliwej transakcji (możliwość akcji w systemie rezerwacyjnym), 'n' lub wartość jak w przypadku checkAvailability().
- [manager] string : email | callcenter | auto - (określa kto obsługuje transakcję),
- [status] string :
new | toAccept | accepted | option | payment | payed | finished | failed | canceled,
- [offerId] TOfferId : identyfikator oferty
- [payer] TPerson : dane płatnika
- [reservationNumber] string : numer rezerwacji w systemie operatora,
- [reservationDate] string : data założenia rezerwacji / opcji,
- [price] string : cena całkowita w groszach,
- [currency] string : waluta np.: PLN, EUR, USD itp ...
- [advance] string : wysokość wymaganej zaliczki / dopłaty w groszach,
- [advanceTo] string : termin płatności zaliczki / dopłaty - data i godzina (YYYY-MM-DD HH:MM:SS),
- [advancesTotal] int : suma wpłaconych zaliczek w groszach,
- [requiredAdvanceUrl] : string link do systemu płatności na kwotę wymaganej zaliczki,
- [requiredFullAdvanceUrl] : string link do systemu płatności na pełną kwotę dopłaty,
- [participants] TParticipants : lista uczestników imprezy,
- [priceParticipants] array of TPriceParticipant : tablica elementów ze składnikami ceny uczestnika. indeksy odpowiadają osobom z tablicy w polu [participants],
- [offerDetails] TOfferDetails : opis oferty, której dotyczy zamówienie,
- [additionalServices] TadditionalServices : lista usług dodatkowych (w przygotowaniu),
- [payments] TPayments : lista płatności powiązanych z transakcją,
- [createDate] string : data utworzenia transakcji (YYYY-MM-DD),
- [createTime] string : godzina utworzenia transakcji (HH:MM:SS),
- [lastModifyDate] string : data ostatniej modyfikacji (YYYY-MM-DD),
- [lastModifyTime] string : godzina ostatniej modyfikacji (HH:MM:SS),

TPriceParticipant

- [totalPrice] int : cena całkowita za uczestnika w groszach,
- [additionalServices] (w przygotowaniu),

TPayment

Struktura zawierająca komplet informacji związanych z płatnościami do transakcji.

Opis pól :

- [status] string : OK | FAIL,
- [value] int : wartość w groszach,
- [info] string : dodatkowa informacja tekstowa,
- [createDate] : data rejestracji płatności w systemie ROOP,
- [createTime] string : godzina utworzenia transakcji (HH:MM:SS),
- [lastModifyDate] string : data ostatniej modyfikacji (YYYY-MM-DD),
- [lastModifyTime] string : godzina ostatniej modyfikacji (HH:MM:SS),

Przykład :

raczej zbędny.

TPayments

Zawiera listę wszystkich płatności powiązanych z rezerwacją. Jest tablicą, której każdy z elementów jest typem TPayment.

Przykład :

raczej zbędny.

Dodatkowe informacje techniczne dla API

Ogólny opis standardów:

- Kodowanie znaków : UTF-8
- Adres serwera API : roop.twojeplatnosci.com.pl
- Adres panelu klienta : <http://twojeplatnosci.com.pl/panel.html>

Wymagania techniczne:

- Obsługa CURL,
- Zainstalowany loader ionCube : <http://www.ioncube.com/>,
- PHP w wersji 5.2 lub nowsza.

Kody błędów.

Kody błędów zwracane są poprzez pola klasy :

- `lastErrorId` : int
- `lastErrorDescription` : string
- **opisy kodów błędów będą dostępne pod stosownym adresem [www : roop.twojeplatnosci.com.pl/errors.csv](http://roop.twojeplatnosci.com.pl/errors.csv)**

(chwilowo jako załącznik do dokumentu)