

OPIS TECHNICZNY

SYSTEM HOSTEDSMS

Wersja 1.8.10
Warszawa, styczeń 2023

SPIS TREŚCI

1. Wprowadzenie	3
2. Podstawowe Parametry systemu HostedSMS	3
Dostępność.....	3
Definicja znaków i długości wiadomości SMS	3
3. Bezpieczeństwo	4
4. Dostępne Interfejsy	4
Interfejs webserwis2SMS	4
Metody Web service	4
Sygnatury metod.....	5
Uwierzytelnienie klienta	9
Parametry metod / Pola struktur Delivery Report / InputSMS.	9
Raport dostarczenia	12
Odebrany SMS.....	13
Interfejs SimpleApi	14
Metoda SendSMS.....	14
Mechanizm AntySpamowy.....	14
Przykłady	15
Interfejs FullApi	17
Uwierzytelnienie	17
Mapowanie Operacji.....	17
Przekazywanie parametru Phone/Phones	18
Przykłady	18

1. WPROWADZENIE

System HostedSMS (w skrócie HSMS) przeznaczony jest do przesyłania hurtowych ilości wiadomości SMS. System HostedSMS zostanie udostępniony Klientowi jako usługa. DCS odpowiedzialny jest za opracowanie oraz jego utrzymanie w infrastrukturze DCS. Rolą DCS jest także zapewnienie odpowiednich łącz do sieci GSM.

2. PODSTAWOWE PARAMETRY SYSTEMU HOSTEDSMS

DOSTĘPNOŚĆ

DCS gwarantuje, że dostępność Systemu HSMS nie będzie mniejsza niż 99,5% w skali miesiąca kalendarzowego. Planowane i uzgodnione z Klientem przerwy techniczne nie są liczone jako niedostępność systemu.

DEFINICJA ZNAKÓW I DŁUGOŚCI WIADOMOŚCI SMS

System HSMS może wysłać wiadomości SMS o maksymalnie długości 4000 znaków. Odpowiada to 27 SMS-om, w przypadku stosowania tylko znaków z alfabetu podstawowego GSM7¹ oraz 60 znaków, w przypadku stosowania pełnego alfabetu UNICODE.

Stosowanie znaków rozszerzonych GSM7 nie powoduje zmiany wiadomości na UNICODE, ale każdy znak z alfabetu rozszerzonego jest kodowany przez dwa znaki alfabetu podstawowego².

- 612 znaków jeśli treść wiadomości zawiera tylko znaki z zakresu GSM7

Liczba znaków w wiadomości SMS	Liczba wiadomości SMS jakie otrzyma odbiorca	Liczba SMSów wykazanych w billingu dla Klient
1-160	1	1
161-306	1	2
307-459	1	3
460-612	1	4
Pow. 612	1 ³	(Liczba znaków/153) - zaokrąglone w górę

- 272 znaki jeśli treść wiadomości zawiera znaki wymagające kodowania UNICODE

¹ Alfabet podstawowy GSM7 oznacza tabelę „Basic Character Set” ze specyfikacji GSM 03.38

² Alfabet rozszerzony GSM7 to znaki $\{ \sim \} \in$ zgodnie z tabelą „Extended Character Set” ze specyfikacji GSM 03.38

³ UWAGA – Nie wszystkie modele telefonów posiadają możliwości składania wiadomości składającej się z większej ilości segmentów. Nie zaleca się stosowania wiadomości dłuższych niż 10 segmentów.

Liczba znaków w wiadomości SMS	Liczba wiadomości SMS jakie otrzyma odbiorca	Liczba SMSów wykazanych w billingu dla Klient
1-70	1	1
71-134	1	2
135-201	1	3
202-268	1	4
Pow. 269	1 ⁴	(Liczba znaków / 67) - zaokrąglone w górę

3. BEZPIECZEŃSTWO

Dostęp do Interfejsu będzie wymagał autoryzacji loginem (adres e-mail) i hasłem. Całość komunikacji pomiędzy DCS i Klient będzie szyfrowana z użyciem certyfikatów wydanych przez zaufane centrum certyfikacji.

Dodatkowo system posiada funkcjonalność tzw. białej listy (White List) adresów IP. Domyślnie funkcjonalność ta jest wyłączona. W celu zabezpieczenia konta należy skonfigurować listę dozwolonych numerów IP w Panelu Klienta (konfiguracja dostępna dla użytkowników z uprawnieniami Administratora) dodając do listy przynajmniej jeden adres IP. Po włączeniu listy dozwolonych numerów IP wszystkie zapytania przychodzące z adresów spoza listy zostaną zablokowane.

4. DOSTĘPNE INTERFEJSY

INTERFEJS WEBSERWIS2SMS

Interfejs Webserwis2SMS przeznaczony jest do zautomatyzowane wysyłek SMSów realizowanych bezpośrednio przez systemy Klienta, jak również umożliwia odbieranie raportów dostarczenia wiadomości SMS.⁵

Interfejs Webserwis2SMS będzie dostępny pod adresami:

- <https://api.hostedsms.pl/WS/SmsSender.asmx> - podstawowy ośrodek przetwarzania DCS
- <https://api2.hostedsms.pl/WS/SmsSender.asmx> - zapasowy ośrodek przetwarzania DCS

METODY WEB SERVICE

SENDSMS - wysyła pojedynczą wiadomość. Metoda zwraca true jeżeli system DCS przyjął wiadomość do wysłania.

⁴ UWAGA – Nie wszystkie modele telefonów posiadają możliwości składania wiadomości składającej się z większej ilości segmentów. Nie zaleca się stosowania wiadomości dłuższych niż 10 segmentów.

⁵ Istnieje również możliwość uruchomienia interfejsu przesyłającego SMS-y wysłane z telefonów komórkowych do klienta. W celu aktywacji dwukierunkowej wersji interfejsu konieczny jest kontakt z DCS.

SENDSMSSES – wysyła wiadomość do grupy telefonów. Metoda zwraca true jeżeli system DCS przyjął wszystkie wiadomości do wysłania. Operacja jest transakcyjna – albo wszystkie wiadomości zostaną przyjęte do wysłania, albo żadna.

GETUNREADDELIVERYREPORTS – zwraca wszystkie nieprzeczytane raporty klienta, jednocześnie zaznaczając je jako przeczytane. Maksymalnie 100 raportów jest zwracanych na raz. Pusta tablica informuje o tym, że nie ma możliwych raportów do pobrania.

GETDELIVERYREPORTS – zwraca grupę raportów dostarczenia wiadomości, dla podanych identyfikatorów wiadomości. Zwracana jest tablica typu DeliveryReport.

GETUNREADINPUTSMSSES – zwraca wszystkie nieprzeczytane odebrane smsy, jednocześnie zaznaczając je jako przeczytane. Maksymalnie 100 sms-ów jest zwracanych na raz. Pusta tablica informuje o tym, że nie ma możliwych sms-ów odebranych do pobrania.

GETINPUTSMSSES – zwraca grupę odebranych sms-ów, dla podanych kryteriów wyszukiwania. Zwracana jest tablica typu InputSms.

GETVALIDSENDERS - zwraca grupę dostępnych nadpisów..

CHECKPHONES- sprawdza poprawność przekazanych numerów. Funkcjonalność ta jest analogiczna do sprawdzania numerów przy ładowaniu pliku do wysyłki – zwraca informację które numery są poprawne, które zablokowane, które są duplikatami numerów a które znajdują się na czarnej liście.

CONVERTTOGSM7- normalizuje tekst do GSM7 usuwając polskie znaki diakrytyczne.

SYGNATURY METOD

```
public bool SendSmses (  
    string UserEmail,  
    string Password,  
    string[] Phones,  
    string Message,  
    string Sender,  
    string TransactionId,  
    DateTime? ValidityPeriod,  
    int Priority,  
    bool FlashSms,  
    string CostCenter,  
    bool? ConvertMessageToGSM7,  
    out string ErrorMessage,  
    out DateTime CurrentTime,
```

```

        out Guid[] MessageIds
    )
public bool SendSms (
    string UserEmail,
    string Password,
    string Phone,
    string Message,
    string Sender,
    string TransactionId,
    DateTime? ValidityPeriod,
    int Priority,
    bool FlashSms,
    string CostCenter,
    bool? ConvertMessageToGSM7,
    out string ErrorMessage,
    out DateTime CurrentTime,
    out Guid? MessageId
)
public bool GetUnreadDeliveryReports (
    string UserEmail,
    string Password,
    out string ErrorMessage,
    out DateTime CurrentTime,
    out DeliveryReport[] DeliveryReports
)
public bool GetDeliveryReports (
    string UserEmail,
    string Password,
    Guid[] MessageIds,

```

```

        bool MarkAsRead,
        out string ErrorMessage,
        out DateTime CurrentTime,
        out DeliveryReport[] DeliveryReports
    )
public bool GetInputSmses (
    string UserEmail,
    string Password,
    DateTime? From,
    DateTime? To,
    string Recipient,
    bool MarkAsRead,
    out string ErrorMessage,
    out DateTime CurrentTime,
    out InputSms[] InputSms
)
public bool GetUnreadInputSmses (
    string UserEmail,
    string Password,
    out string ErrorMessage,
    out DateTime CurrentTime,
    out InputSms[] InputSms
)
public bool GetValidSenders (
    string UserEmail,
    string Password,
    out string[] Senders,
    out string ErrorMessage,
    out DateTime CurrentTime

```

```

    )

public bool CheckPhones(
    string userEmail,
    string Password,
    string[] Phones,
    out string[] ValidPhones,
    out string[] InvalidPhones,
    out string[] Duplicates,
    out string[] BlockedPhones,
    out string ErrorMessage,
    out DateTime CurrentTime
)

public bool ConvertToGsm7(
    string userEmail,
    string Password,
    string Text,
    out string Gsm7Text,
    out string ErrorMessage,
    out DateTime CurrentTime
)

public struct DeliveryReport
{
    public Guid ReportId;
    public string Phone;
    public Guid MessageId;
    public int Status;
    public DateTime DeliveryTime;
}

```



```

public struct InputSms
{
    public Guid MessageId { get; set; }
    public string Phone { get; set; }
    public string Recipient { get; set; }
    public string Message { get; set; }
    public DateTime ReceivedTime { get; set; }
}

```

UWIERZYTELNIENIE KLIENTA

Uwierzytelnienie klienta następuje za pomocą pary userEmail / Password identyfikującej jednoznacznie klienta usługi HSMS.

PARAMETRY METOD / POLA STRUKTUR DELIVERY REPORT / INPUTSMS.

NUMER TELEFONU

```

string Phone,
string[] Phones,
out string[] ValidPhones,
out string[] InvalidPhones,
out string[] Duplicates,
out string[] BlockedPhones

```

Numer telefonu ma być podawany zawsze w postaci międzynarodowej np: 48xxxxxxxxx.
W przypadku masowego wysyłania wiadomości podawana jest tablica numerów telefonów, zgodnie z powyższym formatem.

W przypadku metody CheckPhones – metoda zwraca numery w podziale na prawidłowe (Valid) nieprawidłowe (Invalid), duplikaty (Duplicates) i zablokowane (BlockedPhones)

NADAWCA – NADPIS

```

string Sender,
out string[] Senders

```

Ciąg alfanumeryczny, który zostanie przekazany do operatora, jako nadpis nadawcy. Należy mieć na uwadze, że w różnych sieciach mogą być różne ustawienia, jeżeli chodzi o dozwoloną formę nadpisu.

Listę wszystkich dozwolonych nadpisów przypisanych do klienta zwraca metoda `GetValidSenders`. W przypadku potrzeby korzystania z innych nadpisów konieczny jest kontakt z DCS celem aktywacji nadpisu.

ODBIORCA

`string` Recipient

Numer, na który przyszedł SMS wejściowy.

Może być podany jako parametr do wyszukiwania odebranych SMS-ów.

TREŚĆ WIADOMOŚCI

`string` Message

System automatycznie będzie kodował wiadomości SMS jako GSM7 lub Unicode, w zależności od tego czy w treści znajdowały się jakieś znaki spoza zestawu GSM7

Maksymalna długość wiadomości SMS w systemie HSMS to 4000 znaków.

IDENTYFIKATOR TRANSAKCJI

`string` TransactionId

Unikalny zestaw znaków identyfikujący próbę wysłania SMS-a / grupy SMS-ów.

Kolejne próby z tym samym identyfikatorem:

- Zwróć wartość `true` i te same parametry wyjściowe, jeżeli wywołanie nastąpiło z tymi samymi parametrami.
- Zwróć wartość `false` i komunikat błędu, w przypadku gdyby parametry się różniły.

Maksymalna wielkość to 128 znaków.

CZAS WAŻNOŚCI SMS-A

`DateTime?` ValidityPeriod

Czas ważności SMS-a podana jako chwila w czasie. W przypadku, w którym czas bieżący jest większy niż czas ważności SMS-a, próba wysyłki będzie odrzucona.

System HSMS nie wyśle SMS-a po upływie czasu ważności.

Jeżeli przyjmuje wartość `NULL`, DCS przyjmuje, że czas ważności nie został ustawiony i ważność wiadomości SMS zostanie ustawiona na domyślną dla danej sieci GSM.

Czas ważności musi zostać podany według czasu obowiązującego w Warszawie.

PRIORYTET

`int` Priority

Liczba z przedziału 0-3, gdzie 0 oznacza najniższy priorytet, a 3 najwyższy.

Wiadomość z niższym priorytetem zostanie wysłana jedynie wówczas, jeżeli w systemie nie czekają na wysłanie żadne wiadomości z wyższym priorytetem.

FLASH SMS

`bool` FlashSms

Wartość **Prawda** oznacza, że wiadomość zostanie wysłana jako SMS typu Flash. Sms typu Flash jest wyświetlany na telefonie bezpośrednio po odebraniu i nie jest zapamiętywany w pamięci telefonu.

Ze względu na ograniczenia techniczne standardu GSM wiadomość typu Flash może składać się tylko z jednego segmentu i zawierać jedynie znaki z alfabetu GSM7. Maksymalna długość wiadomości Flash to 160 znaków.

KONWERTUJ WIADOMOŚĆ DO FORMATU GSM7

`bool?` ConvertMessageToGSM7

Wartość **Prawda** oznacza, że wiadomość zostanie przekonwertowana przed wysłaniem do formatu GSM7, czyli zostaną z niej usunięte polskie znaki diakrytyczne i zastąpione swoimi łacińskimi odpowiednikami. Inne symbole spoza alfabetu GSM7 również zostaną usunięte.

W przypadku niepodania wartości lub podania **Fałsz**, nie nastąpi konwersja.

CENTRUM KOSZTÓW

`string` CostCenter

Nazwa centrum kosztów. Aby używać centrum kosztów, najpierw musi zostać skonfigurowane w panelu klienta. Tylko wartości z listy stworzonych centrów kosztów są dozwolone.

W przypadku wyłączonej funkcjonalności, pole powinno przyjmować wartość null.

OZNACZ JAKO PRZECZYTANE

`bool` MarkAsRead

Zaznacz jako przeczytane. W przypadku pobierania raportów dostarczenia lub odebranych SMS-ów dla konkretnych wiadomości, istnieje opcja zaznaczenia ich jako przeczytanie, aby nie były już zwracane przy pomocy metody GetUnreadDeliveryReports.

ZAKRES CZASU

`DateTime?` From,

`DateTime?` To

Parametry do wyszukiwania odebranych SMS-ów.

OPIS BŁĘDU

```
out String ErrorMessage
```

Tekstowy opis błędu, w postaci zrozumiałej dla człowieka, w języku angielskim. W wypadku poprawnej operacji, pole powinno być puste. Wszystkie metody interfejsu zwracają wartość logiczną **Prawda/Falsz**, informującą o prawidłowym zakończeniu operacji. Wartość ErrorMessage ma znaczenie tylko w przypadku, jeżeli metoda zwróciła wartość **Falsz**.

Zmienna wyjściowa.

CZAS BIEŻĄCY

```
out DateTime CurrentTime
```

Celem ominięcia problemów związanych z czasem, system DCS podaje czas według własnego zegara.

Zmienna wyjściowa.

IDENTYFIKATOR WIADOMOŚCI

```
out Guid? MessageId
```

```
out Guid[] MessageId,
```

```
Guid[] MessageIds,
```

Identyfikator wiadomości podany jako Guid, zwracany celem połączenia wiadomości z raportem dostarczenia. Zmienna ma znaczenie wyłącznie, jeżeli metody SendSms lub SendSmses zwracają wartość **Prawda**.

Zmienna wyjściowa dla metod SendSms i SendSmses oraz zmienna wejściowa dla metody GetDeliveryReports.

RAPORT DOSTARCZENIA

```
public struct DeliveryReport
{
    Guid ReportId;
    string Phone;
    Guid MessageId;
    int Status;
    DateTime DeliveryTime;
}
```

Struktura z informacjami zawierająca pojedynczy opis raportu dostarczenia. Znaczenie pól wyjaśniono w pozostałych punktach.

IDENTYFIKATOR RAPORTU DOSTARCZENIA

```
Guid ReportId;
```

Jednoznacznie identyfikuje raport dostarczenia wiadomości.

STATUS DOSTARCZENIA

`int` Status

- 1 wiadomość niedostarczona (finalnie).
- 2 wiadomość niedostarczona z powodu przedawnienia. Wiadomość nie została dostarczona w podanym czasie ValidityPeriod.
- 3 wiadomość została odrzucona przez operatora.
- +1 wiadomość dostarczona.
- 0 stan nieustalony, nie powinien występować.

Jako zasadę można badać warunek >0 i <0 . Kolejne statusy, jeżeli będą dodawane, będą zachowywać zasadę, że ujemne wartości dotyczą niedostarczenia wiadomości, a dodatnie dotyczą dostarczenia wiadomości.

CZAS DOSTARCZENIA.

`DateTime` DeliveryTime

Czas uznania, że SMS został / finalnie nie został dostarczony. Według czasu w Warszawie.

CZAS ODEBRANIA

`DateTime` ReceivedTime

Czas, w którym SMS został zapisany w systemie DCS.

ODEBRANY SMS

```
public struct InputSms
{
    public Guid MessageId { get; set; }
    public string Phone { get; set; }
    public string Recipient { get; set; }
    public string Message { get; set; }
    public DateTime ReceivedTime { get; set; }
}
```

Wszystkie pola zgodne z definicjami podanymi wcześniej.

INTERFEJS SIMPLEAPI

Interfejs SimpleAPI przeznaczony jest dla klientów chcących w możliwie jak najprostszy sposób wysłać wiadomości SMS, niepotrzebujących zaawansowanych możliwości Webservice2SMS.

Interfejs jest dostępny pod adresami:

- <https://api.hostedsms.pl/SimpleApi> - podstawowy ośrodek przetwarzania DCS
- <https://api2.hostedsms.pl/SimpleApi> - zapasowy ośrodek przetwarzania DCS

Interfejs został wykonany w technologii ASP.NET Web API⁶.

METODA SENDSMS

Interfejs oferuje jedną metodę SendSms z następującymi parametrami:

- UserEmail – Login do systemu
- Password – Hasło do systemu
- Sender – Przypisany dla klienta nadpis
- Phone – Numer telefonu odbiorcy
- Message – Treść wiadomości
- v – Opcjonalny parametr do mechanizmu antyspamowego (patrz niżej)
- ConvertMessageToGSM7 – Opcjonalny parametr do konwersji treści wiadomości do formatu GSM7

W odpowiedzi system zwraca jedną z poniższych dwóch wartości:

- MessageId – Identyfikator wiadomości
- ErrorMessage – Komunikat błędu

Znaczenie parametrów jest dokładnie takie samo jak w przypadku interfejsu Webservice2SMS.

Zapytania są wykonywane za pomocą HTTP POST. Formatowanie zapytania i odpowiedzi odbywa się za pomocą nagłówek Content-Type i Accept (patrz przykłady poniżej).

MECHANIZM ANTYSZPAMOWY

Aby ochronić odbiorcę SMS-ów przed zalewem SMS-ów, API wyposażone jest w mechanizm antyspamowy.

Powtórne wywołanie API danego dnia z tymi samymi parametrami spowoduje zwrócenie zapamiętanej odpowiedzi i nie spowoduje wysłania SMS-a.

W przypadku, w którym istnieje potrzeba wysłania drugiego SMS-a o tej samej treści na ten sam numer telefonu, można to zrobić ustawiając opcjonalny parametr „v” na dowolną unikalną wartość.

⁶ Patrz: <http://www.asp.net/web-api>

PRZYKŁADY

Technologia ASP.NET WEB API pozwala na wykonywanie zapytań w różnym formacie. Poniżej przykład dla wywołania w postaci Request – FormData / Response – Json:

PRZYKŁAD – FORMDATA / JSON

Poniższy przykład pokazuje wywołanie serwisu za pomocą formatu Form Data oraz za pomocą formatu JSON.

W obu przypadkach odpowiedź będzie taka sama i zwrócona będzie w formacie JSON.

Wywołanie (FORM DATA):

```
POST https://api.hostedsms.pl/SimpleApi HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

UserEmail=useremail%40dcs.pl&Password=correctpassword&Sender=TEST&Phone=48xxxxxxxx&Message=TEST
```

Wywołanie (JSON):

```
POST https://api.hostedsms.pl/SimpleApi HTTP/1.1
Accept: application/xml
Content-Type: application/json; charset=UTF-8

{"UserEmail":"useremail@dcs.pl","Password":"correctpassword","Sender":"TEST","Phone":"48502505988","Message":"TEST712"}
```

Odpowiedź Pozytywna:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/7.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Tue, 25 Feb 2014 15:59:38 GMT
Content-Length: 52

{"MessageId":"ebc789b2-d111-4e9c-8a0e-d3ae0896364c"}
```

Odpowiedź Negatywna:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/7.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
```

Date: Tue, 25 Feb 2014 15:55:42 GMT
Content-Length: 38

```
{"ErrorMessage": "Invalid Credentials"}
```

PRZYKŁAD - XML

Poniżej odpowiednik poprzedniego przykładu w formacie XML.

Wywołanie:

```
POST https://api.hostedsms.pl/SimpleApi HTTP/1.1
Accept: application/xml
Content-Type: application/xml; charset=UTF-8

<SimpleSms>
  <UserEmail>useremail@dcs.pl</UserEmail>
  <Password>correctpassword</Password>
  <Sender>TEST</Sender>
  <Phone>48xxxxxxxx</Phone>
  <Message>TEST712</Message>
</SimpleSms>
```

UWAGA: W przypadku wywołania XML, kolejność parametrów ma znaczenie.

Odpowiedź Pozytywna:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/xml; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-SourceFiles: =?UTF-8?B?QzpcHJvamVjdHNCRENTLkhvc3RIZFNtc1xEQ1MuSFNNUy5TaW1wbGVBcGk=?=
X-Powered-By: ASP.NET
Date: Mon, 03 Mar 2014 16:06:32 GMT
Content-Length: 138

<MessageSend xmlns:i="http://www.w3.org/2001/XMLSchema-instance"><MessageId>1eba6eb2-888f-41cb-b146-9a1debd8621</MessageId></MessageSend>
```

Odpowiedź Negatywna:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/xml; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-SourceFiles: =?UTF-8?B?QzpcHJvamVjdHNCRENTLkhvc3RIZFNtc1xEQ1MuSFNNUy5TaW1wbGVBcGk=?=
X-Powered-By: ASP.NET
```


Date: Mon, 03 Mar 2014 16:11:55 GMT
Content-Length: 115

```
<Error xmlns:i="http://www.w3.org/2001/XMLSchema-instance"><ErrorMessage>Invalid  
Credentials</ErrorMessage></Error>
```

INTERFEJS FULLAPI

Interfejs FullApi jest wersją REST pełnego interfejsu Web Service. Każda operacja możliwa do wykonania w interfejsie Web Service, może być również wykonana za pomocą interfejsu FullApi. Dla interfejsu jest przygotowany klient w postaci paczki Nuget – HostedSms FullApi Client.

Interfejs FullApi jest też dostępny na platformie Swagger, która umożliwia wygodne korzystanie z niego w przeglądarce.

Interfejs dostępny jest pod adresami:

- <https://api.hostedsms.pl/FullApi/> - podstawowy ośrodek przetwarzania DCS
- <https://api2.hostedsms.pl/FullApi/> - zapasowy ośrodek przetwarzania DCS
- <https://api.hostedsms.pl/FullApi/swagger/ui/index> - interfejs FullApi na platformie Swagger

UWIERZYTELNIENIE

Uwierzytelnienie w FullApi jest typu BASIC.

Oprócz tego serwer również zaakceptuje nagłówek Authorization postaci username:password bez kodowania Base64.

MAPOWANIE OPERACJI

Operacje są mapowane w obiekty Restowe zgodnie z poniższą tabelką:

Operacja	Objekt	Metoda	Uwagi
<i>SendSmses</i> <i>SendSms</i>	<i>Smses</i>	POST	Interfejs posiada tylko jedną metodę do wysyłania SMS-ów. W celu wysłania jednego SMS-a, należy użyć tablicy Phones z jednym elementem.
<i>GetUnreadDeliveryReports</i> <i>GetDeliveryReports</i>	<i>DeliveryReports</i>	GET	W przypadku podania parametrów MessageIds i/lub MarkAsRead, wywoływana jest metoda GetDeliveryReports. W przeciwnym razie GetUnreadDeliveryReports.

GetUnreadInputSmses GetInputSmses	InputSmses	GET	Podanie jakiegokolwiek parametru powoduje wywołanie GetInputSmses. Brak parametrów spowoduje wywołanie GetUnreadInputSmses.
GetValidSenders	ValidSenders	GET	
ConvertToGsm7	Gsm7	POST	
CheckPhones	CheckPhones	POST	

PRZEKAZYWANIE PARAMETRU PHONE/PHONES

W Przypadku używania formatowania XML, należy przekazać tablicę Phones:

```
<Phones>
```

```
<Phone>Telefon1</Phone>
```

```
<Phone>Telefon1</Phone>
```

```
</Phones>
```

W przypadku formatowania Form Data, należy przekazać kolejne telefony w formularzu:

```
Phone=Telefon1&Phone=Telefon2
```

W przypadku JSON-a, należy przekazać parametr Phone (który może być tablicą):

```
"Phone":["Telefon1","Telefon2"]
```

Formaty zapytań i odpowiedzi pokazane są na poniższych przykładach:

PRZYKŁADY

UWIERZYTELNIENIE – FORMDATA/JSON

Zapytanie bez uwierzytelnienia:

```
GET https://api.hostedsms.pl/FullApi/ValidSenders HTTP/1.1
User-Agent: Fiddler
```

Odpowiedź serwera 401:

```
HTTP/1.1 401 Unauthorized
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
```

```
Server: Microsoft-IIS/8.0
WWW-Authenticate: Basic
X-AspNet-Version: 4.0.30319
X-SourceFiles: =?UTF-
8?B?QzpcCHJvamVjdHhNcRENTLkhvc3RIZFNtc1xEQ1MuSFNNUy5GdWxsQXBpXFZhbGlkU2VuZGVycw==?=
X-Powered-By: ASP.NET
Date: Fri, 24 Jul 2015 14:13:56 GMT
Content-Length: 94

{"ErrorMessage":"Invalid Login or Password","CurrentTime":"2015-07-24T16:13:56.5601277+02:00"}
```

Zapytanie z uwierzytelnieniem Basic:

```
GET https://api.hostedsms.pl/FullApi/ValidSenders HTTP/1.1
User-Agent: Fiddler
Authorization: Basic c2R1X2FwaUBkY3MucGw6cXdIMTIzcXdl
```

Odpowiedź serwera:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-SourceFiles: =?UTF-
8?B?QzpcCHJvamVjdHhNcRENTLkhvc3RIZFNtc1xEQ1MuSFNNUy5GdWxsQXBpXFZhbGlkU2VuZGVycw==?=
X-Powered-By: ASP.NET
Date: Fri, 24 Jul 2015 13:17:19 GMT
Content-Length: 30

{"Senders":["TEST","SDUSOFT"]}
```

PRZYKŁADY – XML

Wywołanie – GetValidSenders:

```
GET https://api.hostedsms.pl/FullApi/ValidSenders HTTP/1.1
User-Agent: Fiddler
Authorization: Basic c2R1X2FwaUBkY3MucGw6cXdIMTIzcXdl
Accept: application/xml
Host: localhost:53596
```

Odpowiedź – GetValidSenders:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/xml; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
```

X-SourceFiles: =?UTF-8?B?QzpcchJvamVjdHNCRENTLkhvc3RlZFNtc1xEQ1MuSFNNUy5GdWxsQXBpXFZhbGlkU2VuZGVycw==?=
X-Powered-By: ASP.NET
Date: Fri, 24 Jul 2015 14:16:43 GMT
Content-Length: 205

<ValidSenders xmlns:i="http://www.w3.org/2001/XMLSchema-instance"><CurrentTime>2015-07-24T16:16:43.6298734+02:00</CurrentTime><Senders><Sender>TEST</Sender><Sender>SDUSOFT</Sender></Senders></ValidSenders>

Wywołanie – GetInputSms:

GET <https://api.hostedsms.pl/FullApi/InputSmses?From=2014-07-09+18%3a00%3a00&To=2014-07-09+18%3a00%3a10&MarkAsRead=False> HTTP/1.1
User-Agent: Fiddler
Authorization: Basic c2R1X2FwaUBkY3MucGw6cXdIMTlzcXdl
Accept: application/xml
Host: localhost:53596

Odpowiedź – GetInputSms:

HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/xml; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-SourceFiles: =?UTF-8?B?QzpcchJvamVjdHNCRENTLkhvc3RlZFNtc1xEQ1MuSFNNUy5GdWxsQXBpXElucHV0U21zZXM=?=
X-Powered-By: ASP.NET
Date: Fri, 24 Jul 2015 14:38:59 GMT
Content-Length: 60429

<InputSmsResponse xmlns:i="http://www.w3.org/2001/XMLSchema-instance"><CurrentTime>2015-07-24T16:38:59.5690067+02:00</CurrentTime><InputSmses><InputSms><Message>@É\$¥èéùìòÇØÅ_ÆæßÉ!\#%'()*,-./0123456789;:<=>?¡ABCDEFGHIJKLMN O P Q R S T U V W X Y Z Ä Ö Ñ Ü § ¤ abcdefghijklm

nopqrstuvwxyzäönüa^{}[~]|€ </Message><MessageId>0fc068ca-a0d6-4b94-b5c8-0049694dd5a2</MessageId><Phone>48620000333</Phone><ReceivedTime>2014-07-09T15:48:38.317</ReceivedTime><Recipient>102</Recipient></InputSms><InputSms><Message>@É\$¥èéùìòÇØÅ_ÆæßÉ!\#%'()*,-./0123456789;:<=>?¡ABCDEFGHIJKLMN O P Q R S T U V W X Y Z Ä Ö Ñ Ü § ¤ abcdefghijklm

nopqrstuvwxyzäönüa^{}[~]|€ </Message><MessageId>8315e6a3-b360-416b-9305-03e94408fb51</MessageId><Phone>48620000259</Phone><ReceivedTime>2014-07-09T15:48:32.553</ReceivedTime><Recipient>102</Recipient></InputSms></InputSmsResponse>

Wywołanie – GetDeliveryReports:

GET [https://api.hostedsms.pl/FullApi/DeliveryReports?MessageIds\[\]=ebc789b2-d111-4e9c-8a0e-d3ae0896364c&MessageIds\[\]=69827e59-8bfa-4e39-94b5-b992aabab1e1&MarkAsRead=False](https://api.hostedsms.pl/FullApi/DeliveryReports?MessageIds[]=ebc789b2-d111-4e9c-8a0e-d3ae0896364c&MessageIds[]=69827e59-8bfa-4e39-94b5-b992aabab1e1&MarkAsRead=False) HTTP/1.1
User-Agent: Fiddler
Host: localhost:53596
Authorization: Basic c2R1X2FwaUBkY3MucGw6cXdIMTlzcXdl

Accept: application/xml

Odpowiedź – GetDeliveryReports:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/xml; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-SourceFiles: =?UTF-8?B?QzpcchJvamVjdHNCRENTLkhvc3RIZFNtc1xEQ1MuSFNNUy5GdWxsQXBpXERlbgI2ZXJ5UmVwb3J0cw==?=
X-Powered-By: ASP.NET
Date: Fri, 24 Jul 2015 15:56:02 GMT
Content-Length: 686

<DeliveryReportResponse xmlns:i="http://www.w3.org/2001/XMLSchema-instance"><CurrentTime>2015-07-24T17:56:02.9738596+02:00</CurrentTime><DeliveryReports><DeliveryReport><DeliveryTime>2013-10-03T16:42:21.693</DeliveryTime><MessageId>69827e59-8bfa-4e39-94b5-b992aabab1e1</MessageId><Phone>48502506086</Phone><ReportId>6bd24782-f4c2-49f1-bfea-0000082dc8a4</ReportId><Status>1</Status></DeliveryReport><DeliveryReport><DeliveryTime>2014-02-25T16:00:50.147</DeliveryTime><MessageId>ebc789b2-d111-4e9c-8a0e-d3ae0896364c</MessageId><Phone>48502505988</Phone><ReportId>ef8104a8-2b32-4cb0-942f-5bfb88004beb</ReportId><Status>1</Status></DeliveryReport></DeliveryReports></DeliveryReportResponse>
```

Wywołanie – SendSmses:

```
POST https://api.hostedsms.pl/FullApi/Smses HTTP/1.1
User-Agent: Fiddler
Host: localhost:53596
Authorization: Basic c2R1X2FwaUBkY3MucGw6cXdIMTIzcXdl
Content-Type: application/xml; charset=UTF-8
Accept: application/xml
Content-Length: 256

<FullSms>
<Phones>
<Phone>48502505988</Phone>
</Phones>
<Message>Test</Message>
<Sender>TEST</Sender>
<TransactionId>sdfwavydwfe</TransactionId>
<Priority>1</Priority>
<CostCenter>test</CostCenter>
<FlashSms>true</FlashSms>
</FullSms>
```

Odpowiedź – SendSmses:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/xml; charset=utf-8
```

Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-SourceFiles: =?UTF-8?B?QzpcHJvamVjdHNCRENTLkhvc3RlZFNtc1xEQ1MuSFNNUy5GdWxsQXBpXFNtc2Vz?=
X-Powered-By: ASP.NET
Date: Fri, 24 Jul 2015 16:51:59 GMT
Content-Length: 223

```
<MessageSend xmlns:i="http://www.w3.org/2001/XMLSchema-instance"><CurrentTime>2015-07-24T18:51:59.2935229+02:00</CurrentTime><MessageIds><MessageId>1c80140f-a1fe-45e8-854b-056405a86097</MessageId></MessageIds></MessageSend>
```

Przykład wywołania CheckPhones (bez nagłówków):

```
<CheckPhonesRequest>  
<Phones>  
<Phone>48502505988</Phone>  
<Phone>502505988</Phone>  
<Phone>12333dddwww</Phone></Phones>  
</CheckPhonesRequest>
```

Odpowiedź do CheckPhones (bez nagłówków):

```
<CheckPhonesResponse xmlns:i="http://www.w3.org/2001/XMLSchema-instance">  
<CurrentTime>2020-06-12T09:54:53.4757295+02:00</CurrentTime>  
<ValidPhones>  
<Phone>48502505988</Phone>  
</ValidPhones>  
<InvalidPhones>  
<Phone>12333dddwww</Phone>  
</InvalidPhones>  
<Duplicates>  
<Phone>48502505988</Phone>  
</Duplicates>  
<BlockedPhones/>  
</CheckPhonesResponse>
```

Wywołanie Gsm7 (bez nagłówków):

```
<Gsm7Request>  
<Text>żać</Text>  
</Gsm7Request>
```

Odpowiedź Gsm7 (bez nagłówków):

```
<Gsm7Response xmlns:i="http://www.w3.org/2001/XMLSchema-instance">  
<CurrentTime>2020-06-12T10:41:12.3962847+02:00</CurrentTime>  
<Gsm7Text>zac</Gsm7Text>  
</Gsm7Response>
```

PRZYKŁADY – JSON / FORM DATA

Wywołanie SendSms (Form Data):

```
POST https://api.hostedsms.pl/FullApi/Smses HTTP/1.1
User-Agent: Fiddler
Authorization: Basic c2R1X2FwaUBkY3MucGw6cXdIMTlzcXdl
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Host: localhost:53596
Content-Length: 133

Phone=48502505988&Phone=48502505989&Message=Test&Sender=TEST&TransactionId=sefsfasdwdfgdf3&
Priority=1&FlashSms=true&CostCenter=Test
```

Wywołanie SendSms (JSON):

```
POST https://api.hostedsms.pl/FullApi/Smses HTTP/1.1
User-Agent: Fiddler
Authorization: Basic c2R1X2FwaUBkY3MucGw6cXdIMTlzcXdl
Content-Type: application/json; charset=utf-8
Host: localhost:53596
Content-Length: 161

{"Phone":["48502505988","48502505989"],"Message":"Test","Sender":"TEST","TransactionId":"sefsfasdwdfgdf3","Priority":"1","FlashSms":"true","CostCenter":"Test"}
```

Odpowiedź:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-SourceFiles: =?UTF-8?B?QzpcHJvamVjdHNCRENTLkhvc3RlZFNtc1xEQ1MuSFNNUy5GdWxsQXBpXFNtc2Vz?=?
X-Powered-By: ASP.NET
Date: Mon, 27 Jul 2015 08:37:26 GMT
Content-Length: 144

{"MessageIds":["8ef5a0b2-8bb8-41da-b70a-0753d3ad1609","1eb3c387-cc29-4d78-9c4b-30c8646b1cbe"],"CurrentTime":"2015-07-27T10:37:26.8137627+02:00"}
```