

Dokumentacja WebAPI

Integracja HIS z TOPSOR

Wersja: 1.2

Metryka

Nazwa Projektu: Dokumentacja WebAPI
Nazwa Produktu: Integracja HIS z TOPSOR
Opis: Dokumentacja WebAPI dla integracja HIS z TOPSOR

Druk Plik: Dokumentacja api dla HIS v1.2.odt
Ilość stron: 16
Data modyfikacji: 2021-01-29
Data wydruku: 2009-05-28

Wykonawca: Infobox Sp. z o.o.
Adres Wykonawcy: ul. Kraszewskiego 9, 43-400 Cieszyn
Kontakt: tel.: +48 33 851 06 03 | infolinia: 0 800 122 120
Autorzy: Kacper Skrzyp (KS), Łukasz Szotek (ŁS)

Metryka zmian

Wersja	Data wydania	Autor	Opis zmian	Akcja (*)	Rozdziały
1.0	2020-12-16	KS, ŁS	Pierwsza wersja	WS	W
1.1	2020-12-17	KS, ŁS	Dodanie funkcji wyrejestrowania Uzupełnione projekty graficzne propozycji implementacji w HIS	WS	3.6 5.2
1.2	2021-01-29	ŁS	Poprawki redakcyjne	We	3.5

(*) Akcje: WS=Wstaw, Z=Zamień, We= Weryfikuj, N= Nowy | Rozdziały: W=Wszystkie
Autorzy: patrz metryka dokumentu

Uwagi:

Należy upewnić się czy ten dokument jest ostatnią wersją. Wydruki i lokalne kopie plików mogą stać się nieaktualne w wyniku zmian w dokumencie źródłowym.

Spis treści

1 AUTORYZACJA.....	4
2 DOSTĘP.....	5
2.1 REST.....	5
2.2 SOAP.....	5
3 DOSTĘPNE FUNKCJE.....	6
3.1 WYŚWIETLENIE LISTY PACJENTÓW.....	6
3.1.1 REST.....	6
3.1.2 SOAP.....	6
3.2 REJESTRACJA.....	6
3.2.1 REST.....	6
3.2.2 SOAP.....	7
3.3 KOREKTA REJESTRACJI.....	7
3.3.1 REST.....	7
3.3.2 SOAP.....	8
3.4 WYPISANIE.....	8
3.4.1 REST.....	8
3.4.2 SOAP.....	8
3.5 PRZENIESIENIE DO KOLEJKI.....	8
3.5.1 REST.....	8
3.5.2 SOAP.....	9
3.6 WYREJESTROWANIE.....	9
3.6.1 REST.....	9
3.6.2 SOAP.....	9
4 BŁĘDY.....	10
4.1 REST.....	10
4.1.1 Struktura.....	10
4.1.2 Przykładowy błąd.....	10
4.2 SOAP.....	10
4.2.1 Struktura.....	10
4.3 LISTA KODÓW.....	10
5 DODATKOWE INFORMACJE.....	12
5.1 DOSTĘPNE FLAGI.....	12
5.2 WYGLĄD OKNA W HIS.....	13
5.2.1 Propozycja wyglądu okna do wiązania numerka TOPSOR z rejestrowanym pacjentem.....	13
5.2.2 Propozycja wyglądu okna edycji pacjenta z możliwością dokonania korekty w postaci wyrejestrowania.....	13
5.3 WSDL.....	13

1 Autoryzacja

Autoryzacja poprzez „**HTTP Basic authentication**” czyli umieszczenie w nagłówku requesta parametru "Authorization" którego wartość to tekst "Basic" z zakodowanym loginem i hasłem użytkownika połączonymi znakiem ":".

Przykładowy sposób stworzenia parametru nagłówka

```
Authorization = 'Basic ' + base64_encode('his:123456789');
```

Przykładowy request za pomocą curla:

```
curl --request GET \  
--url 'https://192.168.1.100:447/rest/v1.0/patients' \  
--header 'Authorization: Basic aGlzOjEyMzQ1Njc4OQ==' \  

```

Jeśli dane są nie prawidłowe to serwer zwróci w odpowiedzi kod **401** i jeśli oczekuje json to dodatkowo zwróci następującą treść:

```
{  
  "success": false,  
  "errors": [  
    {  
      "code": 401,  
      "text": "unauthorized"  
    }  
  ]  
}
```

Autoryzacja w metodzie SOAP jest również wymagana przez dodanie w/w requesta w nagłówku. Jeśli dane są nieprawidłowe, serwer zwróci komunikat **wrong auth** w postaci XML.

2 Dostęp

2.1 REST

Wzór linka: **{host}/rest/{wersja}**

gdzie **{host}** to ip z portem do strony a **{wersja}** to wersja api aktualna wersja to **v1.0**

Przykład: <https://192.168.1.100:447/rest/v1.0/>

2.2 SOAP

Wzór linka do wsdl'a: **{host}/soap?wsdl**

gdzie **{host}** to ip z portem do strony

Przykład: <https://192.168.1.100:447/soap?wsdl>

Wzór linka na które są wysyłane zapytania: **{host}/soap**

gdzie **{host}** to ip z portem do strony

Przykład: <https://192.168.1.100:447/soap>

3 Dostępne funkcje

3.1 Wyświetlenie listy pacjentów

3.1.1 REST

URL	{host}/rest/{wersja}/patients	
METODA	GET	
PARAMETRY	BRAK	
ZWRACA		
Tablice pacjentów składających się z	Nazwa	Opis
	uuid	Identyfikator użytkownika
	ticket	Numer biletu
	created	Data i godzina wydrukowania biletu w formacie YYYY-MM-DD hh:mm:ss
Przykładowa odpowiedź:		
<pre>[{ "uuid": "0e421c27-533f-4f0c-8c27-d1161e5e0010", "ticket": "Z023", "created": "2020-09-29 08:11:35" }, { "uuid": "c1274256-d457-4a06-bf3d-47360cef84d4", "ticket": "Z021", "created": "2020-09-28 15:04:15" }, { "uuid": "0a8db9a4-7dd3-4870-9a22-60226ec343ca", "ticket": "Z020", "created": "2020-09-28 15:00:01" },]</pre>		

3.1.2 SOAP

Funkcja	patients
Request	BRAK
Response	ArrayOfPatients

3.2 Rejestracja

3.2.1 REST

URL	{host}/rest/{wersja}/patients/{uuid}/registration		
METODA	POST		
PARAMETRY			
Nazwa	Typ	Wymagane	Opis
uuid	Uuid	Tak	Identyfikator pacjenta

firstName	String	Nie	Imię pacjenta
lastName	String	Nie	Nazwisko pacjenta
pesel	String	Nie	Pesel pacjenta
birthday	Date	Nie	Data urodzenia pacjenta w formacie YYYY-MM-DD
gender	Enum (K,M,N)	Nie	Płeć pacjenta
mainBookNumber	Big int	Nie	Numer książki głównej pacjenta
departmentalBookNumber	Big int	Nie	Numer książki oddziałowej pacjenta
flags	Enum[] (12)	Nie	Identyfikatory flag
firstLook	Bool	Nie	Czy nadać wyższy priorytet
ZWRACA			
<pre>{ "success": true, "errors": [] }</pre>	Nazwa		Opis
	success		Czy zapytanie wykonało się pomyślnie
	errors		Tablica z błędami

3.2.2 SOAP

Funkcja	register
Request	RegisterForm
Response	Result

3.3 Korekta rejestracji

Może zostać przesłany tylko aktualizowany parametr czyli np. uuid + pesel

3.3.1 REST

URL	{host}/rest/{wersja}/patients/{uuid}/registration		
METODA	PATCH		
PARAMETRY			
Nazwa	Typ	Wymagane	Opis
uuid	Uuid	Tak	Identyfikator pacjenta
firstName	String	Nie	Imię pacjenta
lastName	String	Nie	Nazwisko pacjenta
pesel	String	Nie	Pesel pacjenta
birthday	Date	Nie	Data urodzenia pacjenta w formacie YYYY-MM-DD
gender	Enum (K,M,N)	Nie	Płeć pacjenta
mainBookNumber	Big int	Nie	Numer książki głównej pacjenta
departmentalBookNumber	Big int	Nie	Numer książki oddziałowej pacjenta
flags	Enum[] (12)	Nie	Identyfikatory flag
firstLook	Bool	Nie	Czy nadać wyższy priorytet
ZWRACA			

<pre>{ "success": true, "errors": [] }</pre>	Nazwa	Opis
	success	Czy zapytanie wykonało się pomyślnie
	errors	Tablica z błędami

3.3.2 SOAP

Funkcja	correct
Request	RegisterForm
Response	Result

3.4 Wypisanie

3.4.1 REST

URL	{host}/rest/{wersja}/patients/{uuid}		
METODA	DELETE		
PARAMETRY			
Nazwa	Typ	Wymagane	Opis
uuid	Uuid	Tak	Identyfikator pacjenta
ZWRACA			
<pre>{ "success": true, "errors": [] }</pre>	Nazwa	Opis	
	success	Czy zapytanie wykonało się pomyślnie	
	errors	Tablica z błędami	

3.4.2 SOAP

Funkcja	end
Request	uuid(string) - Identyfikator pacjenta
Response	Result

3.5 Przeniesienie do kolejki

3.5.1 REST

URL	{host}/rest/{wersja}/patients/{uuid}		
METODA	PUT		
PARAMETRY			
Nazwa	Typ	Wymagane	Opis
uuid	Uuid	Tak	Identyfikator pacjenta
queueId	Int	Tak	Identyfikator kolejki
queueName	String	Tak	Nazwa kolejki
ZWRACA			
{	Nazwa	Opis	

<pre>"success": true, "errors": [] }</pre>	success	Czy zapytanie wykonało się pomyślnie
	errors	Tablica z błędami

3.5.2 SOAP

Funkcja	move
Request	uuid - Identyfikator pacjenta queueId - Identyfikator kolejki queueName - Nazwa kolejki
Response	Result

3.6 Wyrejestrowanie

Dodatkowa funkcja „serwisowa” umożliwia wyrejestrowanie błędnie powiązanego numerka z pacjentem.

3.6.1 REST

URL	{host}/rest/{wersja}/patients/{uuid}/registration		
METODA	DELETE		
PARAMETRY			
Nazwa	Typ	Wymagane	Opis
uuid	Uuid	Tak	Identyfikator pacjenta
ZWRACA			
<pre>{ "success": true, "errors": [] }</pre>	Nazwa	Opis	
	success	Czy zapytanie wykonało się pomyślnie	
	errors	Tablica z błędami	

3.6.2 SOAP

Funkcja	unregister
Request	uuid(string) - Identyfikator pacjenta
Response	Result

4 Błędy

4.1 REST

4.1.1 Struktura

success - Czy zapytanie wykonało się pomyślnie

errors - Tablica z błędami w której znajdują się obiekty składające się z kodu(code) i tekstu(text)

4.1.2 Przykładowy błąd

```
{
  "success": false,
  "errors": [
    {
      "code": 420,
      "text": "too less data provided"
    }
  ]
}
```

4.2 SOAP

4.2.1 Struktura

```
<xsd:complexType name="Result">
  <xsd:sequence>
    <xsd:element name="success" type="xsd:boolean"/>
    <xsd:element name="errors" type="ns:ArrayOfErrors"/>
  </xsd:sequence>
</xsd:complexType>
```

4.3 Lista kodów

Kod HTTP	Kod	Tekst	Opis
401	401	wrong auth	Brak lub błędne dane autoryzacji
422	420	too less data provided	Trzeba podać więcej danych niż tylko uuid
	421	required uuid	Uuid jest wymagane
	422	invalid uuid	Uuid jest nieprawidłowe
	423	client with this uuid not exist	Pacjent nie istnieje dla podanego uuid
	424	main book number must be numeric	Numer książki głównej musi być liczbą
	425	departmental book number must be numeric	Numer książki oddziałowej musi być liczbą
	426	required queue id	Identyfikator kolejki jest wymagany
	427	queue id must be numeric	Identyfikator kolejki musi być liczbą

	428	required queue name	Nazwa kolejki jest wymagana
	429	queue name must be string	Nazwa kolejki musi być tekstem
500	500	registration failed check configuration	Rejestracja nie powiodła się
	501	end failed	Wypisanie nie powiodło się
	502	move failed	Przeniesienie nie powiodło się
	503	unregister failed	Wyrejestrowanie nie powiodło się

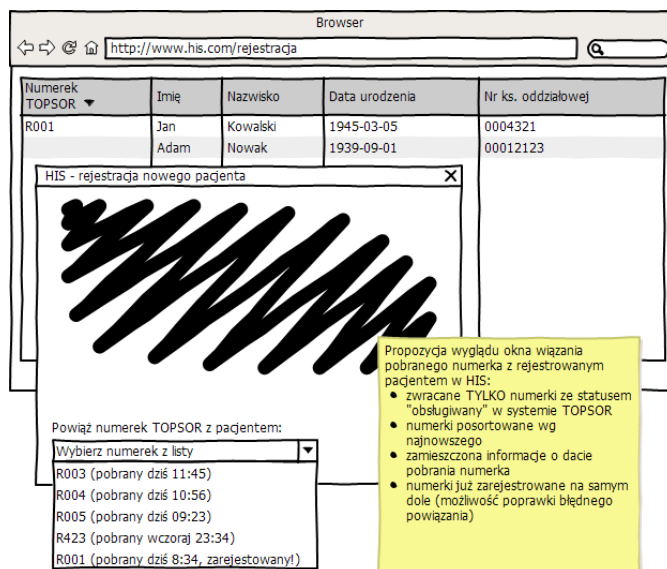
5 Dodatkowe informacje

5.1 Dostępne flagi

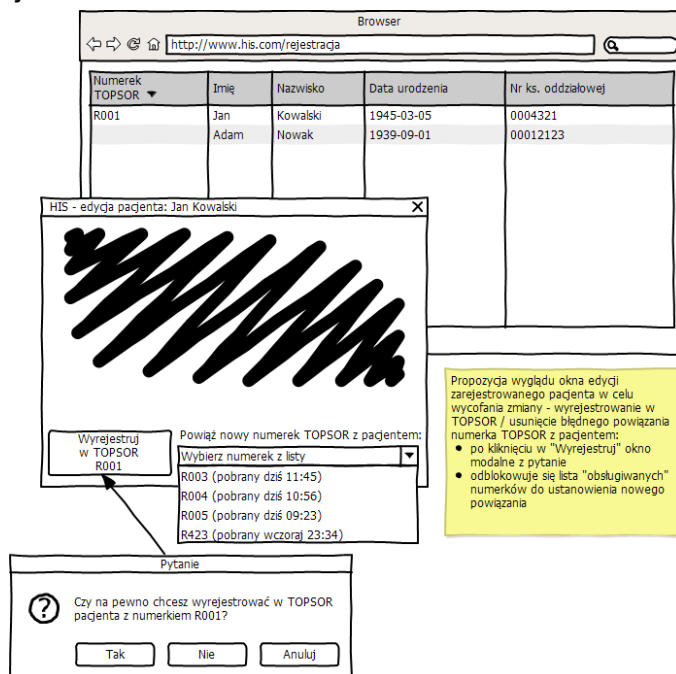
Identyfikator	Opis
1	Kobieta w ciąży
2	Senior
3	Upośledzony ruchowo
4	Upośledzony umysłowo
5	Niewidomy
6	Głuchoniemy
7	Bez kontaktu
8	Pacjent psychiatryczny

5.2 Wygląd okna w HIS

5.2.1 Propozycja wyglądu okna do wiązania numerka TOPSOR z rejestrowanym pacjentem.



5.2.2 Propozycja wyglądu okna edycji pacjenta z możliwością dokonania korekty w postaci wyrejestrowania



5.3 WSDL

```
<?xml version="1.1" encoding="UTF-8"?>  
<definitions name="TopsorService"  
  targetNamespace="http://localhost/topsorservice"
```

```
xmlns:tns="http://localhost/topsorservice"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"  
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"  
xmlns="http://schemas.xmlsoap.org/wsdl/"  
xmlns:ns="http://localhost/topsorservice/types">  
<types>  
  <xsd:schema  
    targetNamespace="http://localhost/topsorservice/types"  
    xmlns="http://localhost/topsorservice/types">  
    <xsd:complexType name="ArrayOfNumbers">  
      <xsd:complexContent>  
        <xsd:restriction base="soapenc:Array">  
          <xsd:attribute ref="soapenc:arrayType" soap:arrayType="xsd:int[]"/>  
        </xsd:restriction>  
      </xsd:complexContent>  
    </xsd:complexType>  
    <xsd:complexType name="ArrayOfPatients">  
      <xsd:complexContent>  
        <xsd:restriction base="soapenc:Array">  
          <xsd:attribute ref="soapenc:arrayType" soap:arrayType="ns:Patient[]"/>  
        </xsd:restriction>  
      </xsd:complexContent>  
    </xsd:complexType>  
    <xsd:element name="Patient" nillable="true" type="ns:Patient"/>  
    <xsd:complexType name="Patient">  
      <xsd:sequence>  
        <xsd:element name="uuid" type="xsd:string"/>  
        <xsd:element name="ticket" type="xsd:string"/>  
        <xsd:element name="created" type="xsd:dateTime"/>  
      </xsd:sequence>  
    </xsd:complexType>  
    <xsd:complexType name="RegisterForm">  
      <xsd:sequence>  
        <xsd:element name="uuid" type="xsd:string"/>  
        <xsd:element minOccurs="0" name="firstName" type="xsd:string"/>  
        <xsd:element minOccurs="0" name="lastName" type="xsd:string"/>  
        <xsd:element minOccurs="0" name="pesel" type="xsd:string"/>  
        <xsd:element minOccurs="0" name="birthday" type="xsd:date"/>  
        <xsd:element minOccurs="0" name="gender" type="xsd:string"/>  
        <xsd:element minOccurs="0" name="mainBookNumber" type="xsd:long"/>  
        <xsd:element minOccurs="0" name="departmentalBookNumber" type="xsd:long"/>  
        <xsd:element minOccurs="0" name="flags" type="ns:ArrayOfNumbers"/>  
        <xsd:element minOccurs="0" name="firstLook" type="xsd:boolean"/>  
      </xsd:sequence>  
    </xsd:complexType>  
    <xsd:element name="Error" nillable="true" type="ns:Error"/>  
    <xsd:complexType name="Error">  
      <xsd:sequence>  
        <xsd:element name="code" type="xsd:int"/>  
        <xsd:element name="text" type="xsd:string"/>  
      </xsd:sequence>  
    </xsd:complexType>  
    <xsd:complexType name="ArrayOfErrors">  
      <xsd:complexContent>  
        <xsd:restriction base="soapenc:Array">  
          <xsd:attribute ref="soapenc:arrayType" soap:arrayType="ns:Error[]"/>  
        </xsd:restriction>  
      </xsd:complexContent>  
    </xsd:complexType>  
    <xsd:element name="Result" nillable="true" type="ns:Result"/>  
    <xsd:complexType name="Result">  
      <xsd:sequence>  
        <xsd:element name="success" type="xsd:boolean"/>  
        <xsd:element name="errors" type="ns:ArrayOfErrors"/>  
      </xsd:sequence>  
    </xsd:complexType>  
  </xsd:schema>  
</types>  
  
<message name="patientsRequest"/>  
  
<message name="patientsResponse">  
  <part name="return" type="ns:ArrayOfPatients"/>  
</message>  
  
<message name="registerRequest">  
  <part name="data" type="ns:RegisterForm"/>  
</message>  
  
<message name="registerResponse">  
  <part name="return" type="ns:Result"/>  
</message>  
  
<message name="correctRequest">
```

```
<part name="data" type="ns:RegisterForm"/>
</message>

<message name="correctResponse">
  <part name="return" type="ns:Result"/>
</message>

<message name="endRequest">
  <part name="uuid" type="xsd:string"/>
</message>

<message name="endResponse">
  <part name="return" type="ns:Result"/>
</message>

<message name="moveRequest">
  <part name="uuid" type="xsd:string"/>
  <part name="queueId" type="xsd:int"/>
  <part name="queueName" type="xsd:string"/>
</message>

<message name="moveResponse">
  <part name="return" type="ns:Result"/>
</message>

<message name="unregisterRequest">
  <part name="uuid" type="xsd:string"/>
</message>

<message name="unregisterResponse">
  <part name="return" type="ns:Result"/>
</message>

<portType name="TopsorServicePortType">
  <operation name="patients">
    <input message="tns:patientsRequest"/>
    <output message="tns:patientsResponse"/>
  </operation>
  <operation name="register">
    <input message="tns:registerRequest"/>
    <output message="tns:registerResponse"/>
  </operation>
  <operation name="correct">
    <input message="tns:correctRequest"/>
    <output message="tns:correctResponse"/>
  </operation>
  <operation name="end">
    <input message="tns:endRequest"/>
    <output message="tns:endResponse"/>
  </operation>
  <operation name="move">
    <input message="tns:moveRequest"/>
    <output message="tns:moveResponse"/>
  </operation>
  <operation name="unregister">
    <input message="tns:unregisterRequest"/>
    <output message="tns:unregisterResponse"/>
  </operation>
</portType>

<binding name="TopsorServiceBinding" type="tns:TopsorServicePortType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="patients">
    <soap:operation soapAction="http://localhost/topsorservice/#patients"/>
    <input>
      <soap:body use="literal" namespace="http://localhost/topsorservice"/>
    </input>
    <output>
      <soap:body use="literal" namespace="http://localhost/topsorservice"/>
    </output>
  </operation>
  <operation name="register">
    <soap:operation soapAction="http://localhost/topsorservice/#register"/>
    <input>
      <soap:body use="literal" namespace="http://localhost/topsorservice"/>
    </input>
    <output>
      <soap:body use="literal" namespace="http://localhost/topsorservice"/>
    </output>
  </operation>
  <operation name="correct">
```

```
<soap:operation soapAction="http://localhost/topsorservice/#correct"/>
  <input>
    <soap:body use="literal" namespace="http://localhost/topsorservice"/>
  </input>
  <output>
    <soap:body use="literal" namespace="http://localhost/topsorservice"/>
  </output>
</operation>
<operation name="end">
  <soap:operation soapAction="http://localhost/topsorservice/#end"/>
  <input>
    <soap:body use="literal" namespace="http://localhost/topsorservice"/>
  </input>
  <output>
    <soap:body use="literal" namespace="http://localhost/topsorservice"/>
  </output>
</operation>
<operation name="move">
  <soap:operation soapAction="http://localhost/topsorservice/#move"/>
  <input>
    <soap:body use="literal" namespace="http://localhost/topsorservice"/>
  </input>
  <output>
    <soap:body use="literal" namespace="http://localhost/topsorservice"/>
  </output>
</operation>
<operation name="unregister">
  <soap:operation soapAction="http://localhost/topsorservice/#unregister"/>
  <input>
    <soap:body use="literal" namespace="http://localhost/topsorservice"/>
  </input>
  <output>
    <soap:body use="literal" namespace="http://localhost/topsorservice"/>
  </output>
</operation>
</binding>
<service name="TopsorServiceService">
  <port name="TopsorServicePort" binding="tns:TopsorServiceBinding">
    <soap:address location="http://localhost/soap"/>
  </port>
</service>
</definitions>
```