



FUNDUSZE EUROPEJSKIE - DLA ROZWOJU WOJEWÓDZTWA PODLASKIEGO

Dotyczy projektu nr WND-RPPD.04.01.00-20-002/11 pn. „Wdrażanie elektronicznych usług dla ludności województwa podlaskiego – część II, administracja samorządowa” realizowanego w ramach Decyzji nr UDA-RPPD.04.01.00-20-002/11-00 z dnia 28.02.2012r.

Wykonanie, dostawa, instalacja, wdrożenie Szyny Danych,
Elektronicznego Zarządzania Dokumentami, Cyfrowego Urzędu oraz
przeprowadzenie szkoleń

Dokumentacja integratora

Autor Dokumentu:	<i>iQ5</i>
Właściciel Dokumentu:	<i>MAXTO</i>
Wersja Dokumentu:	<i>3.0</i>
Status Dokumentu	<i>Draft</i>
Miejsce przechowywania:	<i>Teambox</i>
Data utworzenia:	<i>2014-08-07</i>
Data ostatniej modyfikacji:	<i>2015-03-03 09:39:00</i>
Klient:	<i>Urząd Marszałkowski Województwa Podlaskiego</i>

Historia zmian

Data	Autor	Podsumowanie zmian	Wersja	Zaznaczanie zmian
2014-08-21	iQ5	Utworzenie dokumentu	0.1	Brak
2014-10-14	iQ5	Aktualizacja dokumentu	0.2	Brak

Lista dystrybucyjna

Data	Imię i nazwisko	Rola w procesie / Stanowisko
	Wojciech Gawlikowski	Kierownik projektu MAXTO

Zatwierdzone

Data	Imię i nazwisko	Rola w procesie	Podpis
	Wojciech Gawlikowski	Zatwierdzający	

Spis treści

Wstęp	6
Opis usługi CU SSO	6
Rozpoczęcie integracji	6
Autentykacja użytkownika	6
Krok 1 – Przygotowanie treści żądania autentykacji (AuthRequest).....	6
Krok 2 – Wywołanie żądania autentykacji.....	7
Krok 3 – Przekierowanie do formularza logowania na stronie dostawcy tożsamości	8
Krok 4 - Wysłanie artefaktu SAML do dostawcy usług.....	8
Krok 5 – Wysłanie żądania ArtifactResolve i odebranie odpowiedzi ArtifactResponse	8
Pobranie danych profilu OKI	10
Wylogowanie interesanta	11
Krok 1 Przygotowanie treści żądania wylogowania LogoutRequest	11
Krok 2 Wywołanie żądania wylogowania	11
Adresy usługi	12
Opis usług KAP.WS	13
Pobranie listy jednostek opublikowanych w KAP.....	13
Pobranie szczegółowych danych jednostki	14
Pobranie listy zmian danych jednostek KAP.....	15
Utworzenie nowej jednostki w KAP	16
Aktualizacja danych jednostki w KAP	18
Pobranie listy zdefiniowanych klasyfikacji jednostek	20
Pobranie szczegółów wybranej klasyfikacji (lista kategorii).....	20
Pobranie jednostek należących do wybranej kategorii.....	21
Pobranie kategorii i słów kluczowych przypisanych do jednostki.....	22
Aktualizacja kategorii i słów kluczowych przypisanych do jednostki	23
Rejestracja nowej subskrypcji	23
Anulowanie subskrypcji.....	25
Adres usługi	25
Opis usługi KUP.WS	26
Pobranie listy klasyfikacji usług.....	26
Pobranie szczegółowych danych wybranej klasyfikacji.....	27

Pobranie listy zdefiniowanych opisów usług.....	28
Pobranie listy opublikowanych opisów usług	28
Pobranie listy usług danego podmiotu (lista kart usług).....	28
Pobranie szczegółowych danych wybranej karty usługi	30
Pobranie szczegółowych danych wybranego opisu usługi.....	30
Pobranie szczegółowych danych opublikowanego opisu usługi.....	31
Utworzenie nowego opisu usługi	32
Aktualizacja istniejącego opisu usługi	33
Utworzenie nowej karty usługi.....	34
Aktualizacja istniejącej karty usługi.....	35
Rejestracja nowej subskrypcji	36
Anulowanie subskrypcji.....	38
Adres usługi	39
Opis usługi ESP.WS	39
Przekazanie dokumentu na skrzynkę podawczą jednostki	39
Doręczenie dokumentu systemowi SEOD wybranej jednostki	40
Adres usługi	40
Opis usługi DS.WS.....	40
Doręczenie dokumentów do interesanta w trybie KPA	40
Pobieranie UPD	41
Pobieranie UPD z potwierdzeniem doręczenia	41
Potwierdzenie pobrana UPD przez system zewnętrzny.....	42
Przekazanie dowolnego dokumentu na OKI	42
Doręczenie dokumentów do interesanta w trybie KPA (na potrzeby SEOD).....	42
Przekazanie dowolnego dokumentu na OKI (na potrzeby SEOD)	43
Adres usługi	43
Opis usługi CU.SeodIntegrator	44
Aktualizacja danych sprawy	44
Dodanie płatności.....	44
Aktualizacja płatności.....	44
Dodanie użytkownika wewnętrznego do Cyfrowego Urzędu	45
Aktualizacja użytkownika wewnętrznego Cyfrowego Urzędu	45
Usunięcie użytkownika wewnętrznego Cyfrowego Urzędu.....	45
Resetowanie hasła użytkownika wewnętrznego Cyfrowego Urzędu	Błąd! Nie zdefiniowano zakładki.

Opis złożonych typów danych wykorzystanych w usłudze SeodIntegrator	46
CaseUpdateResponse.....	46
PaymentAddResponse	46
PaymentUpdateResponse	46
UserDto	46
Adres usługi	47

Wstęp

Dokument zawiera szczegółowe informacje w zakresie mechanizmów dostarczanych przez CUWP na potrzeby integracji z systemami zewnętrznymi. W poszczególnych rozdziałach przedstawione zostały zgrupowane tematycznie specyfikacje interfejsów oraz inne niezbędne do integracji systemów informacje.

Opis usługi CU SSO

Usługa Cyfrowego Urzędu będąca dostawcą tożsamości dla interesantów korzystających z systemów wchodzących w skład platformy CUWP. Głównym zadaniem tej usługi jest zapewnienie mechanizmów do realizacji SSO (Single Sign On) w oparciu o protokół SAML 2.0.

Rozpoczęcie integracji

W celu wykonania integracji systemu zewnętrznego z CU SSO, należy przekazać do administratorów Cyfrowego Urzędu adresu URL systemu zewnętrznego w celu zarejestrowania nowego systemu. Zgłaszany system otrzyma dwa certyfikaty: certyfikat z kluczem prywatny, którym będzie podpisywał komunikaty SAML oraz certyfikat publiczny urzędu do weryfikacji otrzymywanych komunikatów.

Autentykacja użytkownika

Poniżej opisane zostały kolejne czynności które należy wykonać w celu przeprowadzenia autentykacji użytkownika. Na każdym kroku przedstawione są przykładowe komunikaty (treść komunikatów zawiera w adresach URL - localhost, które podczas integracji należy zmienić na adresy produkcyjne/testowe usługi SSO. Adresy URL zostały podane w punkcie *Adresy usługi*).

Krok 1 – Przygotowanie treści żądania autentykacji (AuthRequest)

W celu wysłania żądania autentykacji należy utworzyć element AuthRequest zgodny z SAML2.0.

Przykładowe żądanie autentykacji:

```
<samlp:AuthnRequest ID="_BEA435E73572361A9970542BCCF0B9B5" Version="2.0"
IssueInstant="2014-08-22T07:33:49.453Z"
Destination="https://localhost/CU.IdP.Public/SAML/SingleSignOn" ForceAuthn="false"
IsPassive="false" ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
AssertionConsumerServiceURL="https://localhost/CU.Public/SAML/AssertionService?binding
=urn%3aoasis%3anames%3atc%3aSAML%3a2.0%3abindings%3aHTTP-Artifact"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
  <saml:Issuer
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://localhost/CU.Public/</saml:
Issuer>
```

```
<samlp:NameIDPolicy AllowCreate="true" />
</samlp:AuthnRequest>
```

Krok 2 – Wywołanie żądania autentykacji

W celu zainicjowania procesu autentykacji należy przygotować odpowiedni adres URL, a następnie przekierować użytkownika na ten adres.

Adres bazowy: https://adres_serwera/CU.IdP.Public/SAML/SingleSignOn

Do adresu bazowego należy dodać następujące parametry GET:

- binding – typ powiązania. CU SSO obsługuje następujący rodzaj powiązania - **urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect**
- SAMLRequest – treść żądania autentykacji poddana operacji DEFLATE i zakodowana do base64.
- RelayState - identyfikator żądania
- SigAlg - algorytm podpisu. Obsługiwany algorytm:
<http://www.w3.org/2000/09/xmlsig#rsa-sha1>
- Signature – parametr zawierający zakodowany do postaci base64 podpis parametrów żądania (SAMLRequest=wartość&SigAlg=wartość). Podpis wykonywany jest przy pomocy otrzymanego certyfikatu z kluczem prywatnym algorytmem rsa-sha1.

Przykładowe wywołanie:

<https://localhost/CU.IdP.Public/SAML/SingleSignOn?binding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect&SAMLRequest=fZLLbslwEEV%2FJbLUJSTkUYpFqJIAaiTaRjy66KYywYAIY1OPQ9u%2F7yQ8RbdI47FGc%2Bee8bgPbCf3NKnsVk35Z8XBOvkJh%2FpKAmDaNQNoq4f3HeSXq%2FrRaGfZtnYS3tpRJw3bkBoFRO%2F7REnB6h4rsAyZTHldcKW99Dy%2FbnXpUFAw147jIj34gzRQShmG%2BXW2j1Q15W6ZHkrwbrZop2vinZRLaUo3VnyPHFnQm0kn4mNeIXEGWtT8gY3JmsmgdfWBQMQB37JFEZbXWqZCrVCdUwqo6hmIIAqtuNAbUnr3hTR6fJYBPRpPi9aU74ShpeWOAkANzVnphVUO25m3BxEyRfTyT%2Fk19QX9Un1eLKJEeUuYA0MxgYHoy3xqiUYEARPMxZeG7AEu61ZDfa9kwpos7jkb%2B1Pz0AG%2FbqaNjyV%2FrbcnYegQxujdt3r5ofnfb0Bbvlw0JjwY%2BTSKm%2FMsOZxRVZU%2BGG3MFR9vfrDX4B&RelayState=f685eeb-9b44-46d7-b263-478705c6a00f&SigAlg=http:%2F%2Fwww.w3.org%2F2000%2F09%2Fxmlsig%23rsa-sha1&Signature=YRn5CXvaTz7As7wuEFAM9IXsT58VVqgiWqKmmPt2oyD8B6OSRnzyXHoDcqt%2BdTrMmdEUxoYOc5M9j9eU%2F2ugfHEE532rjNG9gmNGr7hU3iiAY9pVCQLbkAIOGg8wthxuUUw87djvV1J33tooyJWdvbmhey%2FXMoOHUvOnnz%2FpRs%3D>

Krok 3 – Przekierowanie do formularza logowania na stronie dostawcy tożsamości

1. Po poprawnym zweryfikowaniu otrzymanego żądania autentykacji, usługa CU SSO przekieruje użytkownika do formularza logowania na stronie dostawcy tożsamości.
2. Użytkownik wprowadza login i hasło i klika Zaloguj.
3. System weryfikuje wprowadzone dane.
4. W przypadku wprowadzenia nieprawidłowych danych system wyświetla stosowny komunikat.
5. W przypadku prawidłowej weryfikacji system wysyła odpowiedź do dostawcy usług.

Krok 4 - Wysłanie artefaktu SAML do dostawcy usług

Na adres zapisany w parametrze AssertionConsumerServiceURL (AuthRequest) wysyłana jest odpowiedź SAML Artifact (przekierowanie użytkownika pod ten adres).

Parametry odpowiedzi:

- SAMLart - zawiera artefakt SAML (wartość tego parametru należy przekazać do ArtifactResolve w kolejnym kroku)
- Binding – typ powiązania, przy użyciu którego została wygenerowana odpowiedź na żądanie autentykacji

Przykładowa odpowiedź SAMLArtifact:

<https://localhost/CU.Public/SAML/AssertionService?binding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact&SAMLart=AAQAABMDYNSeggUNiESlc6r4DEwAAAAAEhcKEVHSCUuFadDTVfiQmAAAAAA%3D&RelayState=f6855eeb-9b44-46d7-b263-478705c6a00f>

Krok 5 – Wysłanie żądania ArtifactResolve i odebranie odpowiedzi ArtifactResponse

Żądanie ArtifactResolve w postaci SOAP należy wysłać na adres:

https://adres_serwera/CU.IdP.Public/SamlArtifactResolve.

Przykładowe żądanie ArtifactResolve:

```
<saml:ArtifactResolve ID="_F0C78953CE02B936759C6A19CB6F0273" Version="2.0"
IssueInstant="2014-08-22T07:38:36.489Z"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:protocol">
  <saml:Issuer
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://localhost/CU.Public/</saml:
Issuer>
```



```

    <samlp:Artifact
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">AAQAABMDYNsEqgUNiES1c6r4DEwAAAAAbgQ
jrUMrIUaWZkHPXiPtTgAAAAA=</samlp:Artifact>
</saml:ArtifactResolve>

```

W rezultacie usługa CU SSO zwróci odpowiedź ArtifactResponse, zawierającą status weryfikacji tożsamości oraz dane konta, dla którego przeprowadzono autentykację. Otrzymanie odpowiedzi ArtifactResponse kończy proces autentykacji użytkownika.

Przykładowa odpowiedź ArtifactResponse:

```

<samlp:ArtifactResponse ID="_5CD8478ABF69759A900CC7E21E836617" Version="2.0"
IssueInstant="2014-08-22T07:40:06.247Z"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
  <saml:Issuer
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">/CU.IdP.Public/</saml:Issuer>
  <samlp:Response ID="_26EF813EDC55776757996BDC245A49F4" Version="2.0"
IssueInstant="2014-08-21T18:45:21.349Z"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <saml:Issuer
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">/CU.IdP.Public/</saml:Issuer>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
      <Reference URI="#_E905681BCCFAA66B6AD7AFCD5CEF5D59">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature" />
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
            <InclusiveNamespaces PrefixList="#default saml ds xs xsi"
xmlns="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transform>
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
        <DigestValue>iy/twB29e/tlYfDXr/Mi2YJrM/U=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>DIRNuM122E8TMKKR5Q2d9p9/SL4p22+Yz/S1sT1N4pD7swQFLpahvnbY65PSB+yPUt65wy

```

```

AlrcxLY4ozUBS3Hs2ST4mTXTGLdEHZvXifQW4owws6NHE0toZrT3yG0h7kHmADXwglpr1s1A8q4sDyMvMqpSM7
eIZQRJbhWBC+EUU=</SignatureValue>
  <KeyInfo>
    <X509Data>

<X509Certificate>MIIB/DCCAWgAwIBAgIQ39LY0FEy9K1A+f1T4C/ELzANBkgqhkig9w0BAQQFADAWMRQwE
gYDVQQDEwtYVwogQ29tcGFueTAeFw0wNDEyMzExNzAwMDBaFw0wOTEyMzExNzAwMDBaMBYxFDASBgNVBAMTC1h
ZWlBDb21wYW55MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC3E55S/VqutRieuJ39TM6HkKh47pC+x3Xk1
Z+gmIPhk2XRbUu0CnJunxnesChjDJ2H01usHoPU2jJbfnffEJRrVw8zDavvVqiye4hHGaSL3i7BDOChzKeQY/8
yifIMFUIK7DOKwfQDUbJf662gac6u0AmNv/CNDIpECWUHokQIDAQBo0swSTBHBgNVHQEEQDA+gBDHk2UyyDjv
EL4gr30aF1NBorGwFjEUMBIGA1UEAxMLWF1aIENvbXBhbnmCEN/S2DhRMvStQPn9U+AvxC8wDQYJKoZIhvcNAQ
EEBQADgYEAIdaguk7RrjeJJtq44DSFatuGtYxASy/MXtdbHhuiYIRNNBgBPB3NWYHVBrZnftBmbHz1Ur61x7ZW
YPqezvKhyKZNgHHkbl0035MHEYNNJhDLdw0QVn4QkZL5MhLHU+8zcaMwTERlQN3rQTAg4paz5oSVDmQyPbUAC/
xsquUP44E=</X509Certificate>
    </X509Data>
  </KeyInfo>
</Signature>
<samlp:Status>
  <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success" />
</samlp:Status>
<saml:Assertion Version="2.0" ID="_49889451345CEC50F77257049D1F2819"
IssueInstant="2014-08-22T07:37:33.982Z"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
  <saml:Subject>
    <saml:NameID>jank</saml:NameID>
  </saml:Subject>
  <saml:AuthnStatement AuthnInstant="2014-08-21T18:45:21.349Z" />
</saml:Assertion>
</samlp:Response>
</samlp:ArtifactResponse>

```

Pobranie danych profilu OKI

W celu pobrania danych profilu interesanta OKI należy postępować analogicznie jak w przypadku autentykacji. Jedyna zmiana dotyczy parametru w adresie podczas wysyłania żądania autentykacji (Krok 2). Do adresu należy dodać parametr `getProfile` i ustawić jego wartość na „true” (`getProfile=true`). Przy tak skonstruowanym żądaniu autentykacji usługa CU SSO zwróci dodatkowe dane profilu interesanta. Dane te znajdą się w elemencie `AttributeStatement` odpowiedzi `ArtifactResponse`.

Gdy żądanie autentykacji jest wykonywane jest w ramach sesji zalogowanego użytkownika Krok 3 jest pomijany.

Przykładowy adres wywołania:

<https://localhost/CU.IdP.Public/SAML/SingleSignOn?binding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect&getProfile=true&SAMLRequest=fZLLbslwEEV%2FJbLUJSTkUYpFqJIAaiTaRiy66KYywYAIY1OPQ9u%2F7yQ8RBdl47FGc%2Bee8bgPbCf3NKnsVk35Z8XBOvwJh%2FpKAmDaNQNoq4f3HeSXq%2FrRaGfZtnYS3tpRJw3bkBoFRO%2F7REnB6h4rsAyZTHldcKW99Dy%2FbnXpUFAw147jIJ34gzRQShmG%2BXW2j1Q15W6ZHkrwbrZop2vinZRLaUo3VnyPHFnQm0kn4mNelXEGWtT8gY3JmsmgdfWBQM QB37JFEZbXWqZCrVcdUwqo6hmIIAqtuNAbUnr3hTR6fjYBPRpPi9aU74ShpeWOAkANzVnphVUO25m3BxEyRfTyT%2Fk19QX9Un1eLKJEEuUYA0MxgYHoy3xqIUYEARPMxZeG7AEu61ZDfa9kwpos7jBk%2B1Pz0AG%2FbqaNjyV%2FrbcnYegQxujdt3r5ofnfb0Bbvlw0JjwY%2BTSK m%2FMsOZxRVZU%2BGG3MFR9vfrDX4B&RelayState=f6855eeb-9b44-46d7-b263-478705c6a00f&SigAlg=http:%2F%2Fwww.w3.org%2F2000%2F09%2Fxmldsig%23rsa-sha1&Signature=YRn5CXvaTz7As7wuEFAM9IXsT58VVqgiWqKmmPt2oyD8B6OSRNzyXHoDcqt%2BdTrMmdEUxoYOc5M9j9eU%2F2ugfHEE532rjNG9gmNGr7hU3iiAY9pVCQLbkAIOGg8wthxuUUw87djvV1J33tooyJWdvmhey%2FXMoOHUvOnnz%2FpRs%3D>

Wylogowanie interesanta

Poniżej opisane zostały kolejne czynności które należy wykonać w celu przeprowadzenia wylogowania użytkownika. Na każdym kroku przedstawione są przykładowe komunikaty (treść komunikatów zawiera w adresach URL - localhost, które podczas integracji należy zmienić na adresy produkcyjne/testowe usługi SSO. Adresy URL zostały podane w punkcie Adresy usług).

Krok 1 Przygotowanie treści żądania wylogowania LogoutRequest

W celu przygotowania treści należy utworzyć dokument XML LogoutRequest zgodny z SAML 2.0.

Przykład treści LogoutRequest:

```
<saml:LogoutRequest ID="_762417B3069BB480D3F0E6FE0DF52AC1" Version="2.0"
IssueInstant="2014-08-22T10:41:43.833Z"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:protocol">
  <saml:Issuer
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://localhost/CU.Public/</saml:
Issuer>
  <saml:NameID xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">jank</saml:NameID>
</saml:LogoutRequest>
```

Krok 2 Wywołanie żądania wylogowania

W celu wylogowania użytkownika należy przygotować odpowiedni adres URL, a następnie przekierować użytkownika na ten adres.

Adres bazowy: https://adres_serwera/CU.IdP.Public/SAML/SingleLogout

Do adresu bazowego należy dodać następujące parametry GET:

- SAMLRequest – treść żądania wylogowania poddana operacji DEFLATE i zakodowana do base64.
- SigAlg - algorytm podpisu. Obsługiwany algorytm: <http://www.w3.org/2000/09/xmlsig#rsa-sha1>
- Signature – parametr zawierający zakodowany do postaci base64 podpis parametrów żądania (SAMLRequest=wartość&SigAlg=wartość). Podpis wykonywany jest przy pomocy otrzymanego certyfikatu z kluczem prywatnym algorytmem rsa-sha1..

Przykład wywołania:

<https://localhost/CU.IdP.Public/SAML/SingleLogout?SAMLRequest=nVlft8lwEP5Xlj7L1m5zYAOLwJhZBGMEfPDF1NHAQtfOXptojP%2B73RhG34xvve%2Fu%2B3GXjoHVgi7VXlnzwF8tB%2BMV2QQ9D5MwJsNZhJOr2Swe4SzK8SLJFzjLL8PpnCDvkWuolJyg0MfIKwAsLyQYJo2DMIkHeDQlww3BNCY0jvxRFD0h760WEmjrOkFWS6oYVEAlqzLQU9L1dLWkTpA2WhIVKoHScRex09d%2FpjMAro2Lh9KDMQ3QIBCqZOKgwATzrX9vX0RVBUgph3jvdOfEiuw%2FTh%2Bo2CFKMEku0FZW7pptLa0QrnZTrTKiqD7yxrIGXaBFzSrRIUruTlXqeL1vlb9UtWvnlQbTk1ZM86PDluwbuj2THLy5WRfZ2Wvz3rg%2B6UKYNgl%2BPU%2B8duazX%2Fy0a9pXv75B%2BgU%3D&SigAlg=http:%2F%2Fwww.w3.org%2F2000%2F09%2Fxmldsig%23rsa-sha1&Signature=VBrzu8A01bHyYGRnK3uq9iEAN5ybpAL6ISqV8SKLq6NPuZAMnWDMkt5FXzmxVRHxhmrMG7b61KIW093rdQABO4hhsAfbo%2FleNtvauFJ28UIFuSAIBHsuvjuxwNmdjB5sTVVfo0vQNh5saZqnKi%2BTKy0qgg7GEdwuZWAjtJD1%2BJ0%3D>

Po otrzymaniu żądania wylogowania usługa CU SSO usuwa kontekst zalogowanego użytkownika i wyświetla stronę z informacją o pomyślnym wylogowaniu użytkownika.

Adresy usługi

- Platforma testowa:
 - Adres, na który wysłane jest żądanie AuthRequest: <https://cutest.wrotapodlasia.pl/CU.IdP.Public/SAML/SingleSignOn>
 - Adres na który wysyłane jest żądanie ArtifactResolve: <https://cutest.wrotapodlasia.pl/CU.IdP.Public/SamlArtifactResolve>
 - Adres, na który wysyłane jest żądanie LogoutRequest <https://cutest.wrotapodlasia.pl/CU.IdP.Public/SAML/SingleLogout>
- Platforma produkcyjna (tymczasowo – do czasu udostępnienia publicznego CU):
 - Adres, na który wysłane jest żądanie AuthRequest: <https://cuprod.wrotapodlasia.pl/CU.IdP.Public/SAML/SingleSignOn>
 - Adres na który wysyłane jest żądanie ArtifactResolve: <https://cuprod.wrotapodlasia.pl/CU.IdP.Public/SamlArtifactResolve>
 - Adres, na który wysyłane jest żądanie LogoutRequest

<https://cuprod.wrotapodlasia.pl/CU.IdP.Public/SAML/SingleLogout>

- Platforma produkcyjna (docelowo – od momentu udostępnienia publicznego CU):
 - Adres, na który wysłane jest żądanie AuthRequest:
<https://cu.wrotapodlasia.pl/CU.IdP.Public/SAML/SingleSignOn>
 - Adres na który wysyłane jest żądanie ArtifactResolve:
<https://cu.wrotapodlasia.pl/CU.IdP.Public/SamlArtifactResolve>
 - Adres, na który wysyłane jest żądanie LogoutRequest
<https://cu.wrotapodlasia.pl/CU.IdP.Public/SAML/SingleLogout>

Opis usług KAP.WS

Usługa ta zapewnia interfejs umożliwiający pobieranie oraz zarządzanie danymi Katalogu Administracji Publicznej Cyfrowego Urzędu Wrót Podlasia (KAP). W szczególności za pomocą interfejsu możliwe jest:

- Pobranie listy jednostek opublikowanych w KAP
- Pobranie szczegółowych danych jednostki
- Pobranie listy zmian danych jednostek KAP
- Utworzenie nowej jednostki w KAP
- Aktualizacja danych jednostki w KAP
- Pobranie listy zdefiniowanych klasyfikacji jednostek
- Pobranie szczegółów wybranej klasyfikacji (lista kategorii)
- Pobranie jednostek należących do wybranej kategorii
- Pobranie kategorii i słów kluczowych przypisanych do jednostki
- Aktualizacja kategorii i słów kluczowych przypisanych do jednostki

Dostęp do usługi zabezpieczony jest poprzez zastosowanie standardu WS-Security. Komunikacja zabezpieczona jest poprzez zastosowania uwierzytelniania na podstawie certyfikatu oraz zastosowania protokołu SSL. Aby uzyskać dostęp do systemu należy zgłosić się do administratora systemu CUWP w celu utworzenia konta systemowego oraz uzyskania niezbędnych certyfikatów (certyfikat identyfikujący klienta usługi oraz certyfikat serwera) i uprawnień umożliwiających zarządzanie danymi jednostek KAP.

Pobranie listy jednostek opublikowanych w KAP

Metoda umożliwia pobranie listy jednostek opublikowanych (widocznych publicznie) w KAP CUWP. W wyniku wywołania metody zwracana jest lista zawierająca podstawowe informacje o jednostkach. W wywołaniu można określić filtr nazwy jednostki, co spowoduje zwrócenie jednostek, które w nazwie zawierają wpisaną frazę.

Nazwa metody: GetUnitList

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
nameFilter	string	Nie	Fragment nazwy jednostki

Zwracany wynik: tablica obiektów UnitInfo

Opis klas:

UnitInfo

Pole	Typ	Opis
Id	int	Identyfikator jednostki w KAP CUWP
ParentUnitId	Nullable<int>	Identyfikator jednostki nadrzędnej w KAP CUWP (tylko jeśli taka jednostka istnieje)
Name	string	Nazwa podmiotu
ShortName	string	Skrócona nazwa podmiotu
NIP	string	NIP
REGON	string	REGON
City	string	Adres jednostki - miejscowość
PostCode	string	Adres jednostki – kod pocztowy
Street	string	Adres jednostki - ulica
Building	string	Adres jednostki – numer budynku
Appartment	string	Adres jednostki – numer lokalu

Pobranie szczegółowych danych jednostki

Metoda umożliwia pobranie szczegółowych danych o jednostce publicznej znajdującej się w KAP CUWP. W wyniku operacji zwracane są szczegółowe dane jednostki oraz struktury organizacyjnej jednostki. Pobranie szczegółowych danych możliwe jest na podstawie identyfikatora jednostki w KAP CUWP, numeru NIP lub numeru REGON. Służą do tego odpowiednie metody wymienione poniżej.

Nazwa metody: GetUnitDetails

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
unitId	int	Tak	Identyfikator jednostki w KAP CUWP

Nazwa metody: GetUnitDetailsByNIP

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
nip	string	Tak	Numer NIP jednostki

Nazwa metody: GetUnitDetailsByREGON**Parametry:**

Nazwa parametru	Typ	Czy wymagany?	Opis
regon	string	Tak	Numer REGON jednostki

Zwracany wynik: string zawierający XML ze szczegółowymi danymi jednostki. Schemat XSD znajduje się w załączniku CuKapUnit.xsd dołączonym do dokumentu.

Możliwe błędy:

- `UnitNotExistsFaultException` – błąd oznaczający brak jednostki o podanym identyfikatorze w KAP CUWP.
- `InvalidParametersFaultException` – błąd oznaczający podanie nieprawidłowych parametrów wywołania metody (np. brak NIP lub REGON)

Pobranie listy zmian danych jednostek KAP

Metoda umożliwia pobranie listy zmian, które zaszły w Katalogu Administracji Publicznej od wskazanej daty. W wyniku wywołania metody zwracana jest informacja o dodaniu, modyfikacji lub usunięciu jednostek KAP.

Nazwa metody: GetUnitChanges**Parametry:**

Nazwa parametru	Typ	Czy wymagany?	Opis
fromDate	datetime	Tak	Data początkowe, od której zwracane są zmiany
filterUnitId	Nullable<int>	Nie	Identyfikator jednostki. Umożliwia odfiltrowanie zmian do konkretnej jednostki.

Zwracany wynik: tablica obiektów UnitChangeInfo

Opis klas:

UnitChangeInfo

Pole	Typ	Opis
ChangeType	UnitChangeType	Rodzaj zmiany. Dostępne wartości: - UnitCreate – utworzenie nowej jednostki - UnitEdit – modyfikacja danych jednostki - UnitDelete – usunięcie jednostki
ChangeDate	datetime	Data wprowadzenia zmiany
UnitId	int	Identyfikator jednostki, której dotyczy zmiana
UnitName	string	Nazwa jednostki, której dotyczy zmiana

Utworzenie nowej jednostki w KAP

Funkcja umożliwia dodanie nowej jednostki do Katalogu Administracji Publicznej CUWP. W wyniku wywołania metody system zweryfikuje przekazane parametry i doda nową jednostkę lub zwróci informacje o stwierdzonych błędach. Po prawidłowym wykonaniu metody w zależności od konfiguracji KAP CUWP dane jednostki będą wymagały zatwierdzenia przez uprawnionego użytkownika przed opublikowaniem lub zostaną automatycznie opublikowane. Do wykonania metody konieczne jest posiadanie uprawnień do tworzenia jednostek KAP przez API.

Nazwa metody: CreateUnit

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
unitXML	string	Tak	XML zawierający dane jednostki. XML musi być zgodny ze schematem XSD jednostki KAP CUWP.
logo	UnitLogoImage	Nie	Dane obrazka z logo jednostki. Parametr opcjonalny umożliwiający dodanie obrazka z logotypem urzędu.
classification	UnitClassificationInfo	Nie	Informacje o kategoriach i słowach kluczowych przypisanych do jednostki. Parametr

			opcjonalny.
requestedPublishDate	Nullable<datetime>	Nie	Parametr opcjonalny umożliwiający określenie przyszłej daty publikacji.

Zwracany wynik: CreateUnitResult

Opis klas:

UnitLogoImage

Pole	Typ	Opis
ImageData	Byte[]	Dane binarne obrazka
ContentType	string	Typ MIME obrazka. Prawidłowe wartości to „image/jpeg”, „image/png” oraz „image/gif”

UnitClassificationInfo

Pole	Typ	Opis
Keywords	string[]	Tablica słów kluczowych przypisanych do jednostki
Categories	ClassificationCategoryInfo	Kategorie/Klasyfikacje przypisane do jednostki. Muszą to być kategorie/klasyfikacje zdefiniowane w KAP CUWP.

ClassificationCategoryInfo

Pole	Typ	Opis
Classification	string	Nazwa klasyfikacji jednostek
Category	string	Nazwa kategorii w ramach wybranej klasyfikacji

CreateUnitResult

Pole	Typ	Opis
Success	bool	Wynik operacji – jeżeli True to operacja została pomyślnie wykonana.
Changeld	int	Identyfikator zmiany w KAP CUWP

ChangeStatus	UnitChangeStatus	Status zmiany: - <i>WaitForApproval</i> – zmiana oczekuje na akceptację - <i>PendingPublish</i> – zmiana została zaakceptowana i oczekuje na publikację (status występuje w przypadku określenia przyszłej daty publikacji) - <i>Published</i> – zmiana została opublikowana - <i>Rejected</i> – zmiana została odrzucona
UnitId	Nullable<int>	Identyfikator utworzonej jednostki. Identyfikator jest zwracany przy natychmiastowej publikacji zmian. W przypadku publikacji zmian w przyszłości lub konieczności zatwierdzenia publikacji przez użytkownika systemu CU identyfikator nie jest zwracany, gdyż jest nadawany w momencie publikacji nowej jednostki.

Możliwe błędy:

- *ValidationFaultException* – błąd oznaczający stwierdzenie błędów walidacyjnych przekazanych parametrów (np. podane dane XML jednostki są niezgodne ze schematem XSD lub podane kategorie/klasyfikacje nie istnieją w KAP CUWP).
- *InvalidParametersFaultException* – błąd oznaczający podanie nieprawidłowych parametrów wywołania metody (np. brak danych XML jednostki)

Aktualizacja danych jednostki w KAP

Funkcja umożliwia aktualizację danych jednostki w KAP CUWP. W wywołaniu metody należy podać XML z kompletnymi danymi jednostki (w tym dane struktury organizacyjnej). Po weryfikacji przekazanych parametrów system zwróci błędy walidacyjne lub zaktualizuje dane jednostki (natychmiast lub z określoną datą). W zależności od konfiguracji KAP CUWP publikacja zmian może wymagać zatwierdzenia operacji przez uprawnionego użytkownika. Do wykonania operacji konieczne jest posiadanie uprawnień do modyfikacji danych jednostki.

Nazwa metody: UpdateUnit

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
unitId	Int	Tak	Identyfikator jednostki w KAP CUWP
unitXML	string	Tak	XML zawierający dane jednostki. XML musi być zgodny ze schematem XSD jednostki KAP CUWP.

logoChange	UnitLogoImageChangeInfo	Nie	Model danych umożliwiający zmianę lub usunięcie logotypu jednostki. Jeżeli NULL, to logo pozostaje bez zmian.
requestedPublishDate	Nullable<datetime>	Nie	Parametr opcjonalny umożliwiający określenie przyszłej daty publikacji zmian.

Zwracany wynik: UpdateUnitResult

Opis klas:

UnitLogoImageChangeInfo

Pole	Typ	Opis
ChangeType	LogoImageChangeType	Rodzaj zmiany: <ul style="list-style-type: none"> - <i>None</i> – logo pozostaje bez zmian - <i>Change</i> – zmiana logo na nowy obrazek - <i>Remove</i> – logo zostanie usunięte
LogoImage	UnitLogoImage	Dane nowego obrazka z logo jednostki. Pole musi być wypełnione w przypadku wybrania rodzaju zmiany <i>Change</i> .

UpdateUnitResult

Pole	Typ	Opis
Success	bool	Wynik operacji – jeżeli True to operacja została pomyślnie wykonana.
ChangeId	int	Identyfikator zmiany w KAP CUWP
ChangeStatus	UnitChangeStatus	Status zmiany: <ul style="list-style-type: none"> - <i>WaitForApproval</i> – zmiana oczekuje na akceptację - <i>PendingPublish</i> – zmiana została zaakceptowana i oczekuje na publikację (status występuje w przypadku określenia przyszłej daty publikacji) - <i>Published</i> – zmiana została opublikowana - <i>Rejected</i> – zmiana została odrzucona

Możliwe błędy:

- `UnitNotExistsFaultException` – błąd oznaczający brak jednostki o podanym identyfikatorze w KAP CUWP
- `ValidationFaultException` – błąd oznaczający stwierdzenie błędów walidacyjnych przekazanych parametrów (np. podane dane XML jednostki są niezgodne ze schematem XSD lub podane kategorie/klasyfikacje nie istnieją w KAP CUWP).
- `InvalidParametersFaultException` – błąd oznaczający podanie nieprawidłowych parametrów wywołania metody (np. brak danych XML jednostki)

Pobranie listy zdefiniowanych klasyfikacji jednostek

Metoda umożliwia pobranie listy klasyfikacji opublikowanych w KAP CUWP.

Nazwa metody: `GetClassificationList`

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
<code>nameFilter</code>	<code>string</code>	Nie	Fragment nazwy klasyfikacji umożliwiający odfiltrowanie zwracanych elementów

Zwracany wynik: tablica obiektów `ClassificationInfo`.

Opis klas:

`ClassificationInfo`

Pole	Typ	Opis
<code>Name</code>	<code>string</code>	Nazwa klasyfikacji
<code>Description</code>	<code>string</code>	Opis klasyfikacji

Pobranie szczegółów wybranej klasyfikacji (lista kategorii)

Metoda umożliwia pobranie informacji o kategoriach zdefiniowanych w ramach wybranej klasyfikacji jednostek. Kategorie mogą być przypisywane do poszczególnych jednostek.

Nazwa metody: `GetClassification`

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
<code>nazwa</code>	<code>string</code>	Nie	Fragment nazwy klasyfikacji

Zwracany wynik: ClassificationDetails.

Opis klas:

ClassificationDetails

Pole	Typ	Opis
Name	string	Nazwa klasyfikacji
Description	string	Opis klasyfikacji
Categories	CategoryInfo[]	Lista kategorii zdefiniowanych w wybranej klasyfikacji

CategoryInfo

Pole	Typ	Opis
Name	string	Nazwa kategorii
Description	string	Opis kategorii

Możliwe błędy:

- ClassificationNotExistsFaultException – błąd oznaczający podanie nazwy klasyfikacji, która nie istnieje w KAP CUWP.
- InvalidParametersFaultException – błąd w sytuacji, gdy nie podano parametru z nazwą klasyfikacji

Pobranie jednostek należących do wybranej kategorii

Metoda umożliwi pobranie listy jednostek KAP, które zostały przypisane do wybranej kategorii klasyfikacji jednostek.

Nazwa metody: GetUnitsByCategory

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
classificationName	string	Tak	Nazwa klasyfikacji
categoryName	String	Tak	Nazwa kategorii

Zwracany wynik: tablica obiektów UnitInfo. (typ został opisany przy metodzie pobierającej listę jednostek)

Możliwe błędy:

- `ClassificationNotExistsFaultException` – błąd oznaczający podanie nazwy klasyfikacji, która nie istnieje w KAP CUWP.
- `CategoryNotExistsFaultException` – błąd oznaczający podanie nazwy kategorii, która nie istnieje we wskazanej klasyfikacji KAP CUWP

Pobranie kategorii i słów kluczowych przypisanych do jednostki

Metoda umożliwia pobranie informacji o słowach kluczowych przypisanych do jednostki oraz kategoriach/klasyfikacjach do których należy jednostka.

Nazwa metody: `GetUnitClassification`

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
<code>unitId</code>	<code>int</code>	Tak	Identyfikator jednostki w KAP CUWP

Zwracany wynik: `UnitClassificationInfoResult`

Opis klas:

`UnitClassificationInfoResult`

Pole	Typ	Opis
<code>UnitId</code>	<code>int</code>	Identyfikator jednostki
<code>UnitName</code>	<code>string</code>	Nazwa jednostki
<code>Keywords</code>	<code>string[]</code>	Tablica słów kluczowych przypisanych do jednostki
<code>Categories</code>	<code>ClassificationCategoryInfo</code>	Kategorie/Klasyfikacje przypisane do jednostki. Muszą to być kategorie/klasyfikacje zdefiniowane w KAP CUWP.

Możliwe błędy:

- `UnitNotExistsFaultException` – błąd oznaczający brak jednostki o podanym identyfikatorze w KAP CUWP

Aktualizacja kategorii i słów kluczowych przypisanych do jednostki

Metoda umożliwia aktualizację słów kluczowych opisujących jednostkę oraz kategorii/klasyfikacji do których przypisana ma być jednostka. Do wykonania operacji konieczne jest posiadanie uprawnień do modyfikacji danych jednostki.

Nazwa metody: UpdateUnitClassification

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
unitId	int	Tak	Identyfikator jednostki w KAP CUWP
classification	UnitClassificationInfo	Tak	Informacje o kategoriach i słowach kluczowych przypisanych do jednostki.

Zwracany wynik: brak

Możliwe błędy:

- `UnitNotExistsFaultException` – błąd oznaczający brak jednostki o podanym identyfikatorze w KAP CUWP
- `InvalidParametersFaultException` – błąd oznaczający podanie nieprawidłowych parametrów wywołania metody (np. brak danych klasyfikacji)
- `CategoryNotExistsFaultException` – błąd oznaczający podanie w danych klasyfikacji kategorii, która nie istnieje w KAP CUWP

Rejestracja nowej subskrypcji

Metoda umożliwia dodanie nowej subskrypcji zmian w KAP CUWP. W ramach parametrów subskrypcji możliwe jest określenie rodzaju zmian, o których system rejestrujący subskrypcje ma być informowany. Możliwe jest otrzymywanie powiadomień o:

- Dodaniu/Opublikowaniu nowej jednostki w KAP CUWP
- Modyfikacji danych jednostki w KAP CUWP (wszystkich lub wybranych jednostek)
- Usunięciu jednostki z KAP CUWP

Podczas rejestracji subskrypcji konieczne jest określenie sposobu informowania o zmianach w KAP CUWP. Dostępne są opcje wysyłania powiadomień na `WebService` lub adres e-mail. Wysyłanie powiadomień na `WebService` wymaga implementacji usługi `WebService` klienta subskrypcji zgodnej z dołączonym do niniejszej dokumentacji plikiem WSDL (plik *KapSubscriptionClient.wsdl*). Komunikacja z usługą klienta subskrypcji jest szyfrowana i odbywa się przy pomocy protokołu HTTPS. Wybór opcji e-mail spowoduje wysyłanie na podany adres wiadomości e-mail zawierającej identyfikator i nazwę subskrypcji oraz plik z informacjami o zmianach w KAP.

Przesyłane informacje o zmianach w KAP zawierając kompletne dane jednostek odzwierciedlające stan jednostki po wprowadzeniu zmiany.

Nazwa metody: RegisterSubscription

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
subscriptionParams	SubscriptionParams	Tak	Parametry subskrypcji

Zwracany wynik: RegisterSubscriptionResult

Opis klas:

SubscriptionParams

Pole	Typ	Opis
Name	string	Nazwa subskrypcji
CreateNotification	bool	Czy wysyłać powiadomienia o dodaniu nowej jednostki
DeleteNotification	bool	Czy wysyłać powiadomienia o usunięciu jednostek
ModifyNotification	bool	Czy wysyłać powiadomienia o modyfikacji danych jednostek
ModifyUnitIds	Int[]	Jeżeli NULL lub pusta tablica to wysyłane są powiadomienia o modyfikacji dowolnej jednostki. Jeśli określono co najmniej jeden identyfikator jednostki to wysyłane są jedynie powiadomienia dotyczące modyfikacji danych wskazanych jednostek
DestinationType	SubscriptionDestinationType	Sposób wysyłki powiadomień. Dostępne wartości: - Email – wysyłanie powiadomień na podany adres e-mail. Należy podać wartość pola Email. - WsAddress – wysyłanie powiadomień WebService o podanym adresie. Należy podać wartość pola WsAddress wskazującą adres usługi WebService do odbioru powiadomień.

Email	string	Adres e-mail
WsAddress	string	Adres usługi WebService zgodnej z WSDL dołączonym do dokumentacji.

RegisterSubscriptionResult

Pole	Typ	Opis
Success	bool	Wynik operacji. True – subskrypcja została pomyślnie zarejestrowana. False – wystąpiły błędy podczas rejestracji subskrypcji, szczegóły błędów zapisane są w polu <i>ErrorMessage</i> .
SubscriptionId	int	Identyfikator utworzonej subskrypcji
ErrorMessage	string	Informacja o błędach podczas rejestracji subskrypcji

Anulowanie subskrypcji

Funkcja umożliwia anulowanie wcześniej zarejestrowanej subskrypcji zmian w KAP CUWP. Po wyłączeniu subskrypcji nie będą wysyłane informacje o zmianach w KAP CUWP.

Nazwa metody: CancelSubscription

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
subscriptionId	int	Tak	Identyfikator subskrypcji otrzymanej podczas rejestracji subskrypcji

Zwracany wynik: wartość typu Boolean. True - subskrypcja została usunięta, False - subskrypcja nie została usunięta

Możliwe błędy:

- *AccessDeniedFaultException* – błąd oznaczający brak uprawnień do anulowania subskrypcji. Anulowanie subskrypcji możliwe jest przez system, który zarejestrował subskrypcję.
- *ValidationFaultException* – błąd oznaczający błędy walidacji przekazanych parametrów (np. podano nieistniejący identyfikator subskrypcji)

Adres usługi

- Platforma testowa:

- Usługa pobierania danych i zarządzania jednostkami KAP:
<https://cutest.wrotapodlasia.pl/CU.WS.KAP/KapService.svc>
- Usługa subskrypcji:
<https://cutest.wrotapodlasia.pl/CU.WS.KAP/SubscriptionService.svc>
- Platforma produkcyjna (tymczasowo – do czasu udostępnienia publicznego CU):
 - Usługa pobierania danych i zarządzania jednostkami KAP:
<https://cuprod.wrotapodlasia.pl/CU.WS.KAP/KapService.svc>
 - Usługa subskrypcji:
<https://cuprod.wrotapodlasia.pl/CU.WS.KAP/SubscriptionService.svc>
- Platforma produkcyjna (docelowo – od momentu udostępnienia publicznego CU):
 - Usługa pobierania danych i zarządzania jednostkami KAP:
<https://cu.wrotapodlasia.pl/CU.WS.KAP/KapService.svc>
 - Usługa subskrypcji:
<https://cu.wrotapodlasia.pl/CU.WS.KAP/SubscriptionService.svc>

Opis usługi KUP.WS

Usługa ta zapewnia interfejs umożliwiający pobieranie oraz zarządzanie danymi Katalogu Usług Publicznych Cyfrowego Urzędu Wrot Podlasia. W szczególności za pomocą interfejsu możliwe jest:

- Pobranie listy klasyfikacji usług
- Pobranie szczegółowych danych wybranej klasyfikacji
- Pobranie listy zdefiniowanych opisów usług
- Pobranie listy usług danego podmiotu (lista kart usług)
- Pobranie szczegółowych danych wybranej karty usługi
- Pobranie szczegółowych danych wybranego opisu usługi
- Utworzenie nowego opisu usługi
- Aktualizacja istniejącego opisu usługi
- Utworzenie nowej karty usługi
- Aktualizacja istniejącej karty usługi

Dostęp do usługi zabezpieczony jest poprzez zastosowanie standardu WS-Security. Komunikacja zabezpieczona jest poprzez zastosowania uwierzytelniania na podstawie certyfikatu oraz zastosowania protokołu SSL. Aby uzyskać dostęp do systemu należy zgłosić się do administratora systemu CUWP w celu utworzenia konta systemowego oraz uzyskania niezbędnych certyfikatów – certyfikat identyfikujący klienta usługi oraz certyfikat serwera. Dostęp do funkcji modyfikujących dane opisów i kart usług wymaga nadania niezbędnych uprawnień dla utworzonego konta systemowego.

Pobranie listy klasyfikacji usług

Metoda umożliwiająca pobranie listy nazw klasyfikacji usług udostępnianych przez KUP CUWP, które zawierają ciąg podany w parametrze metody. W wyniku wywołania metody zwracany jest string zawierający XML z listą nazw klasyfikacji.

Nazwa metody: `dajListeKlasyfikacji`

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
nazwa	string	Nie	Fragment nazwy klasyfikacji

Zwracany wynik: string zawierający XML z listą nazw klasyfikacji. Poniżej widoczna jest postać przykładowego wyniku wywołania metody.

```
<?xml version="1.0" encoding="UTF-8"?>
<Klasyfikacje>
  <NazwaKlasyfikacji>Klasyfikacja terytorialna</NazwaKlasyfikacji>
  <NazwaKlasyfikacji>Zdarzenia życiowe</NazwaKlasyfikacji>
</Klasyfikacje>
```

Pobranie szczegółowych danych wybranej klasyfikacji

Metoda umożliwia pobranie szczegółowych danych klasyfikacji usług o nazwie podanej w parametrze wywołania metody. W wyniku wykonania operacji zwracany jest XML z pozycjami klasyfikacji tworzącymi hierarchiczną strukturę kategorii.

Nazwa metody: `dajListeKlasyfikacji`

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
nazwa	string	Tak	Dokładna nazwa klasyfikacji

Zwracany wynik: string zawierający XML z danymi wybranej klasyfikacji. Poniżej widoczna jest postać przykładowego wyniku wywołania metody.

```
<?xml version="1.0" encoding="UTF-8"?>
<Klasyfikacje>
  <Klasyfikacja>
    <NazwaKlasyfikacji>Zdarzenia życiowe</NazwaKlasyfikacji>
    <PozycjeKlasyfikacji>
      <PozycjaKlasyfikacji>
        <NazwaPozycji>Praca i zatrudnienie</NazwaPozycji>
        <KodPozycji>4</KodPozycji>
        <KodPozycjiNadrzednej></KodPozycjiNadrzednej>
      </PozycjaKlasyfikacji>
      <PozycjaKlasyfikacji>
        <NazwaPozycji>Edukacja</NazwaPozycji>
        <KodPozycji>6</KodPozycji>
        <KodPozycjiNadrzednej></KodPozycjiNadrzednej>
      </PozycjaKlasyfikacji>
      <PozycjaKlasyfikacji>
        <NazwaPozycji>Poszukiwanie pracy</NazwaPozycji>
        <KodPozycji>7</KodPozycji>
        <KodPozycjiNadrzednej>4</KodPozycjiNadrzednej>
      </PozycjaKlasyfikacji>
    </PozycjeKlasyfikacji>
  </Klasyfikacja>
</Klasyfikacje>
```

</Klasyfikacje>

Pobranie listy zdefiniowanych opisów usług

Metoda umożliwia pobranie listy wszystkich zdefiniowanych w KUP CUWP opisów usług (nie koniecznie opublikowanych). W odróżnieniu od metody udostępnianej przez ePUAP zwracane są wszystkie, a nie tylko opublikowane opisy usług. W zwracanych danych dostępna jest informacja o statusie opisu.

Uwaga: Opisy usług w KUP CUWP podlegają wersjonowaniu. W wyniku wykonania operacji zwracana jest lista najnowszych wersji opisów usług oraz ich status publikacji. Aby porać listę opublikowanych opisów usług należy użyć metodę *dajListeOpublikowanychOpisowUslug*.

Nazwa metody: *dajListeOpisowUslug*

Parametry: brak

Zwracany wynik: tablica obiektów *OpisUslugiInfo*

Opis klas:

OpisUslugiInfo

Pole	Typ	Opis
id	int	Identyfikator opisu usługi w systemie CUWP
nazwa	String	Nazwa usługi w systemie CUWP
czyAktualny	bool	Czy opis usługi jest opublikowany

Pobranie listy opublikowanych opisów usług

Metoda umożliwia pobranie listy wszystkich opublikowanych w KUP CUWP opisów usług.

Uwaga: Opisy usług w KUP CUWP podlegają wersjonowaniu. Aby porać listę najnowszych wersji opisów usług należy użyć metodę *dajListeOpisowUslug*.

Nazwa metody: *dajListeOpublikowanychOpisowUslug*

Parametry: brak

Zwracany wynik: tablica obiektów *OpisUslugiInfo*

Pobranie listy usług danego podmiotu (lista kart usług)

Metoda zwracająca listę kart usług danego podmiotu zarejestrowanych Katalogu Usług Publicznych CUWP. W wywołaniu metody należy podać identyfikator podmiotu (nazwę skróconą) z Katalogu Administracji Publicznej CUWP. W wyniku wywołania metody zwrócona zostanie lista zawierająca podstawowe informacje o usługach świadczonych przez wybrany podmiot, dane kart usług oraz formularzy powiązanych z kartą lub opisem usługi.

Uwaga: Karty usług w KUP CUWP podlegają wersjonowaniu. W wyniku wykonania operacji zwracana jest lista najnowszych wersji kart usług oraz ich status publikacji.

Nazwa metody: `dajListeOUslugPodmiotu`

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
<code>idPodmiotu</code>	<code>string</code>	Tak	Dokładna nazwa klasyfikacji

Zwracany wynik: tablica obiektów `UslugPodmiotuInfo`

Opis klas:

`UslugPodmiotuInfo`

Pole	Typ	Opis
<code>idOpisuUslugi</code>	<code>int</code>	Identyfikator opisu usługi w systemie CUWP
<code>nazwaOpisuUslugi</code>	<code>String</code>	Nazwa usługi w systemie CUWP
<code>kartyUslugi</code>	<code>KartaUslugiInfo[]</code>	Tablica zawierająca dane karty usługi. Powinna być dokładnie jedna karta – jeden podmiot może mieć tylko jedną kartę usługi dla wskazanego opisu.

`UslugPodmiotuInfo`

Pole	Typ	Opis
<code>idKartyUslugi</code>	<code>int</code>	Id karty usługi
<code>czyAktualna</code>	<code>bool</code>	Czy karta usługi jest opublikowana
<code>kartaXml</code>	<code>Byte[]</code>	Dane karty w formacie XML zgodnym z formatem karty usługi na platformie ePUAP
<code>formularze</code>	<code>FormularzInfo[]</code>	Informacje o formularzach powiązanych z kartą usługi.

`FormularzInfo`

Pole	Typ	Opis
<code>id</code>	<code>int</code>	Stała wartość 0. Identyfikator nie jest używany w KUP CU.

		Pole pozostało na potrzeby zgodności z formatem ePUAP.
tytul	int	Tytuł/Nazwa formularza
url	bool	Url do formularza zewnętrznego (jeśli czyZewnetrzny ma wartość TRUE) lub przestrzeń nazw formularza opublikowanego w CUWP.
czyZewnetrzny	Byte[]	Czy formularz jest formularzem zewnętrznym, jeśli FALSE to jest formularzem CUWP

Pobranie szczegółowych danych wybranej karty usługi

Metoda zwraca szczegółowe dane wybranej karty usługi. W wywołaniu metody użytkownik podaje identyfikator podmiotu (nazwa skrócona podmiotu w KAP CUWP), identyfikator karty usługi, a system po weryfikacji przekazanych parametrów zwraca dane najnowszej wersji karty usługi.

Nazwa metody: `dajKarteUslugiPodmiotu`

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
idPodmiotu	string	Tak	Identyfikator opisu usługi w KUP CUWP
idKarty	int	TAK	Identyfikator karty usługi

Zwracany wynik: obiekt typu `UsługaPodmiotuInfo` (klasa została opisana we wcześniejszych metodach).

Pobranie szczegółowych danych wybranego opisu usługi

Metoda zwraca szczegółowe dane wybranego opisu usługi. W wywołaniu metody użytkownik podaje identyfikator opisu usługi, a system po weryfikacji przekazanych parametrów zwraca dane najnowszej wersji opisu usługi.

Uwaga: Opisy usług w KUP CUWP podlegają wersjonowaniu. W wyniku wykonania operacji zwracana jest najnowsza wersja opisu usługi (nie koniecznie opublikowana). Aby pobrać dane aktualnie opublikowanego opisu usługi należy wywołać metodę `dajOpublikowanyOpisUslugi`.

Nazwa metody: `dajOpisUslugi`

Parametry:

Nazwa parametru	Typ	Czy	Opis
-----------------	-----	-----	------

		wymagany?	
idUslugi	int	Tak	Identyfikator opisu usługi w KUP CUWP

Zwracany wynik: obiekt typu OpisUslugiSzczegoly

Opis klas:

OpisUslugiSzczegoly

Pole	Typ	Opis
id	int	Identyfikator opisu usługi w systemie CUWP
nazwa	String	Nazwa usługi w systemie CUWP
czyAktualna	bool	Czy opis jest opublikowany
OpisUslugiXml	Byte[]	XML z danymi opisu usługi. XML opisu usługi jest zgodny ze schematem opisu usługi na platformie ePUAP
formularze	FormularzInfo []	Tablica zawierająca dane informacje o formularzach powiązanych z opisem usługi (klasa została opisana we wcześniejszych metodach).

Pobranie szczegółowych danych opublikowanego opisu usługi

Metoda zwraca szczegółowe dane opublikowanej wersji wybranego opisu usługi. W wywołaniu metody użytkownik podaje identyfikator opisu usługi, a system po weryfikacji przekazanych parametrów zwraca dane opublikowanej wersji opisu usługi.

Uwaga: Opisy usług w KUP CUWP podlegają wersjonowaniu. W wyniku wykonania operacji zwracana jest opublikowana wersja opisu usługi. Aby pobrać dane najnowszej wersji opisu usługi należy wywołać metodę *dajOpisUslugi*.

Nazwa metody: *dajOpublikowanyOpisUslugi*

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
idUslugi	int	Tak	Identyfikator opisu usługi w KUP CUWP

Zwracany wynik: obiekt typu OpisUslugiSzczegoly

Utworzenie nowego opisu usługi

Metoda umożliwi dodanie nowego opisu usługi w KUP CUWP. W wyniku wywołania metody system przeprowadzi weryfikację przekazanych parametrów, a następnie zapisze w systemie nowy opis usługi. W zależności od aktualnych ustawień systemu CUWP utworzony opis usługi będzie wymagał akceptacji w celu opublikowania lub zostanie automatycznie opublikowany. Wykonanie operacji wymaga posiadania uprawnień do modyfikacji opisów usług.

Nazwa metody: utworzOpisUslugi

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
xmlOpisUslugi	Byte[]	Tak	Xml zawierający dane o opisie usługi. Xml musi być zgodny ze schematem opisu usługi ePUAP.
xmlKlasyfikacjaUslugi	Byte[]	Nie	Plik XML z informacja o klasyfikacjach/kategoriach oraz słowach kluczowych, które mają być przypisane do tworzonego opisu usługi
czyAktualna	bool	Tak	Dowolna wartość. Parametr nie jest używany. Dodając usługę jest ona automatycznie publikowana lub wymaga akceptacji przed publikacją.
formularze	FormularzInfo[]	Nie	Dane formularzy powiązanych z opisem usługi. Jeżeli NULL to z opisem usługi nie będą powiązane formularze.

Zwracany wynik: identyfikator utworzonego opisu usługi

Plik XML przekazywany w parametrze *xmlKlasyfikacjaUslugi* powinien mieć następujący format:

```
<?xml version="1.0" encoding="UTF-8"?>
<Klasyfikacje>
  <SlovaKluczowe>slowo nr 1, slowo nr 2, test</SlovaKluczowe>
  <Klasyfikacja>
    <NazwaKlasyfikacji>Zdarzenia życiowe</NazwaKlasyfikacji>
    <PozycjeKlasyfikacji>
      <PozycjaKlasyfikacji>
        <NazwaPozycji>Edukacja</NazwaPozycji>
        <KodPozycji>6</KodPozycji>
      </PozycjaKlasyfikacji>
      <PozycjaKlasyfikacji>
        <NazwaPozycji>Praca i zatrudnienie</NazwaPozycji>
        <KodPozycji>4</KodPozycji>
      </PozycjaKlasyfikacji>
    </PozycjeKlasyfikacji>
  </Klasyfikacja>
</Klasyfikacje>
```



```

</Klasyfikacja>
<Klasyfikacja>
  <NazwaKlasyfikacji>Klasyfikacja terytorialna</NazwaKlasyfikacji>
  <PozycjeKlasyfikacji>
    <PozycjaKlasyfikacji>
      <NazwaPozycji>Podlaskie</NazwaPozycji>
      <KodPozycji>44</KodPozycji>
    </PozycjaKlasyfikacji>
  </PozycjeKlasyfikacji>
</Klasyfikacja>
</Klasyfikacje>

```

Możliwe błędy:

- `InvalidParametersFaultException` – błąd oznaczający przekazanie nieprawidłowych parametrów w wywołaniu metody
- `ServiceClassificationFaultException` – błąd występujący w sytuacji, gdy w wywołaniu podano niewłaściwe dane dotyczące klasyfikacji usługi, np. nieistniejącą kategorię lub klasyfikację usług.
- `ValidationFaultException` – błąd walidacji parametrów (np. XML jest niezgodny ze schematem)
- `DuplicateServiceNameFaultException` - błąd występujący w przypadku gdy istnieje już opis usługi o podanej nazwie.

Aktualizacja istniejącego opisu usługi

Metoda umożliwia aktualizację danych opisu usługi w systemie CUWP. Przy użyciu tej metody możliwa jest aktualizacja opisu usługi, formularzy powiązanych z opisem usługi, informacji o klasyfikacji wybranego opisu usługi oraz statusu publikacji. Wykonanie operacji wymaga posiadania uprawnień do modyfikacji opisów usług.

Nazwa metody: `aktualizujOpisUslugi`

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
<code>idOpisUslugi</code>	<code>int</code>	Tak	Identyfikator opisu usługi
<code>xmlOpisUslugi</code>	<code>Byte[]</code>	Nie	Xml zawierający dane o opisie usługi. Xml musi być zgodny ze schematem opisu usługi ePUAP. Jeżeli NULL to dane opisu usługi nie są modyfikowane
<code>xmlKlasyfikacjaUslugi</code>	<code>Byte[]</code>	Nie	Plik XML z informacja o klasyfikacjach/kategoriach oraz słowach kluczowych, które mają być przypisane do tworzonego opisu usługi. Jeżeli NULL to dane w zakresie klasyfikacji nie są modyfikowane.

czyAktualna	bool	Nie	Jeżeli TRUE, to usługa zostaje opublikowana lub zgłoszona do publikacji i wymaga akceptacji (w zależności od konfiguracji CUWP). Jeżeli FALSE to wycofywana jest publikacja opisu usługi. Podanie wartości NULL pozostawia status publikacji bez zmian.
formularze	FormularzInfo[]	Nie	Dane formularzy powiązanych z opisem usługi. Jeżeli NULL to lista formularzy powiązanych z opisem usługi nie jest modyfikowana.

Zwracany wynik: Status operacji (1 - zaktualizowano, -1 - nie odnaleziono usługi)

Możliwe błędy:

- `InvalidParametersFaultException` – błąd oznaczający przekazanie nieprawidłowych parametrów w wywołaniu metody
- `ServiceClassificationFaultException` – błąd występujący w sytuacji, gdy w wywołaniu podano niewłaściwe dane dotyczące klasyfikacji usługi, np. nieistniejącą kategorię lub klasyfikację usług.
- `ValidationFaultException` – błąd walidacji parametrów (np. XML jest niezgodny ze schematem)
- `DuplicateServiceNameFaultException` - błąd występujący w przypadku gdy istnieje już opis usługi o podanej nazwie.

Utworzenie nowej karty usługi

Metoda umożliwia dodanie nowej karty usługi podmiotu dla wskazanego opisu usługi. W wyniku wykonania metody, po pomyślnej weryfikacji przekazanych danych, utworzona zostanie nowa karta usługi. W zależności od ustawień sytemu CUWP karta zostanie automatycznie opublikowana lub publikacja będzie wymagała zatwierdzenia przez uprawnionego użytkownika. Wykonanie operacji wymaga posiadania uprawnień do modyfikacji kart usług jednostki, dla której dodawana jest karta.

Nazwa metody: `utworzKarteUslugi`

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
idPodmiotu	String	Tak	Identyfikator podmiotu (nazwa skrócona) w KAP CUWP

idOpisUslugi	int	Tak	Identyfikator opisu usługi
xmlKartyUslugi	Byte[]	Tak	Xml zawierający dane o opisie usługi. Xml musi być zgodny ze schematem karty usługi ePUAP.
czyAktualna	bool	Tak	Dowolna wartość. Parametr nie jest używany. Dodając kartę jest ona automatycznie publikowana lub wymaga akceptacji przed publikacją.
formularze	FormularzInfo[]	Nie	Dane formularzy powiązanych z kartą usługi. Jeżeli NULL lub pusta tablica to z kartą będą powiązane formularze zdefiniowane dla opisu usługi.

Zwracany wynik: identyfikator utworzonej karty usługi

Możliwe błędy:

- `InvalidParametersFaultException` – błąd oznaczający przekazanie nieprawidłowych parametrów w wywołaniu metody
- `ServiceNotExistsFaultException` – brak opisu usługi o podanym identyfikatorze w systemie
- `ValidationFaultException` – błąd walidacji parametrów (np. XML jest niezgodny ze schematem)
- `DuplicateCardFaultException` - błąd występujący w przypadku gdy istnieje już karta usługi podanego podmiotu dla wybranego opisu usługi.

Aktualizacja istniejącej karty usługi

Metoda umożliwia aktualizację danych karty usługi w systemie CUWP. Przy użyciu tej metody możliwa jest aktualizacja karty usługi, formularzy powiązanych z kartą usługi oraz statusu publikacji karty. Wykonanie operacji wymaga posiadania uprawnień do modyfikacji kart usług jednostki.

Nazwa metody: `aktualizujKarteUslugi`

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
idKartyUslugi	int	Tak	Identyfikator karty usługi która ma być zaktualizowana
idOpisuUslugi	Nullable<int>	Nie	Identyfikator opisu usługi.
xmlKartyUslugi	Byte[]	Nie	Xml zawierający dane o opisie usługi. Xml musi być zgodny ze schematem

			karty usługi ePUAP.
czyAktualna	bool	Nie	Jeżeli TRUE, to usługa zostaje opublikowana lub zgłoszona do publikacji i wymaga akceptacji (w zależności od konfiguracji CUWP). Jeżeli FALSE to wycofywana jest publikacja opisu usługi. Podanie wartości NULL pozostawia status publikacji bez zmian.
formularze	FormularzInfo[]	Nie	Dane formularzy powiązanych z kartą usługi. Jeżeli NULL to lista powiązanych formularzy nie jest modyfikowana.

Zwracany wynik: brak

Możliwe błędy:

- `InvalidParametersFaultException` – błąd oznaczający przekazanie nieprawidłowych parametrów w wywołaniu metody
- `CardNotExistsFaultException` – brak karty o podanym identyfikatorze w systemie
- `ValidationFaultException` – błąd walidacji parametrów (np. XML jest niezgodny ze schematem)

Rejestracja nowej subskrypcji

Metoda umożliwia dodanie nowej subskrypcji zmian w KUP CUWP. W ramach parametrów subskrypcji możliwe jest określenie rodzaju zmian, o których system rejestrujący subskrypcje ma być informowany. Możliwe jest otrzymywanie powiadomień o:

- Dodaniu/Opublikowaniu nowego opisu usługi
- Modyfikacji opisu usługi (publikacji nowej wersji)
- Usunięciu lub wycofaniu publikacji opisu usługi
- Modyfikacji klasyfikacji usług

Podczas rejestracji subskrypcji konieczne jest określenie sposobu informowania o zmianach w KUP CUWP. Dostępne są opcje wysyłania powiadomień na WebService lub adres e-mail. Wysyłanie powiadomień na WebService wymaga implementacji usługi WebService klienta subskrypcji zgodnej z dołączonym do niniejszej dokumentacji plikiem WSDL (plik *KupSubscriptionClient.wsdl*). Komunikacja z usługą klienta subskrypcji jest szyfrowana i odbywa się przy pomocy protokołu HTTPS. Wybór opcji e-mail spowoduje wysyłanie na podany adres wiadomości e-mail zawierającej identyfikator i nazwę subskrypcji oraz plik z informacjami o zmianach w KUP.

Wysyłane powiadomienia zawierają jedynie podstawowe informacje o zmianach w KAP. Pobranie kompletnych danych opisów i klasyfikacji usług możliwe jest przy użyciu pozostałych metod API KUP.

Nazwa metody: RegisterSubscription

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
subscriptionParams	SubscriptionParams	Tak	Parametry subskrypcji

Zwracany wynik: RegisterSubscriptionResult**Opis klas:**

SubscriptionParams

Pole	Typ	Opis
Name	string	Nazwa subskrypcji
ServiceCreate	bool	Czy wysyłać powiadomienia o opublikowaniu nowego opisu usługi
ServiceDelete	bool	Czy wysyłać powiadomienia o wycofaniu publikacji lub usunięciu opisów usług
ServiceChange	bool	Czy wysyłać powiadomienia o modyfikacji opisów usług (publikacji nowej wersji opisu usługi)
ModifyUnitIds	Int[]	Jeżeli NULL lub pusta tablica to wysyłane są powiadomienia o modyfikacji dowolnego opisu usługi. Jeśli określono co najmniej jeden identyfikator opisu usługi to wysyłane są jedynie powiadomienia dotyczące modyfikacji danych wskazanych opisów
ClassificationChange	bool	Czy wysyłać powiadomienia o zmianach definicji klasyfikacji usług? Jeżeli opcja jest włączona to wysyłane będą powiadomienia o dodaniu, modyfikacji oraz usunięciu klasyfikacji i kategorii w ramach poszczególnych klasyfikacji.
DestinationType	SubscriptionDestinationType	Sposób wysyłki powiadomień. Dostępne wartości: - Email – wysyłanie powiadomień na podany adres e-mail. Należy podać wartość pola Email. - WsAddress – wysyłanie powiadomień WebService o podanym adresie. Należy

		podać wartość pola WsAddress wskazującą adres usługi Webservice do odbioru powiadomień.
Email	string	Adres e-mail
WsAddress	string	Adres usługi Webservice zgodnej z WSDL dołączonym do dokumentacji.

RegisterSubscriptionResult

Pole	Typ	Opis
Success	bool	Wynik operacji. True – subskrypcja została pomyślnie zarejestrowana. False – wystąpiły błędy podczas rejestracji subskrypcji, szczegóły błędów zapisane są w polu <i>ErrorMessage</i> .
SubscriptionId	int	Identyfikator utworzonej subskrypcji
ErrorMessage	string	Informacja o błędach podczas rejestracji subskrypcji

Anulowanie subskrypcji

Funkcja umożliwia anulowanie wcześniej zarejestrowanej subskrypcji zmian w KAP CUWP. Po wyłączeniu subskrypcji nie będą wysyłane informacje o zmianach w KAP CUWP.

Nazwa metody: CancelSubscription

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
subscriptionId	int	Tak	Identyfikator subskrypcji otrzymanej podczas rejestracji subskrypcji

Zwracany wynik: wartość typu Boolean. True - subskrypcja została usunięta, False - subskrypcja nie została usunięta

Możliwe błędy:

- `AccessDeniedFaultException` – błąd oznaczający brak uprawnień do anulowania subskrypcji. Anulowanie subskrypcji możliwe jest przez system, który zarejestrował subskrypcję.
- `ValidationFaultException` – błąd oznaczający błędy walidacji przekazanych parametrów (np. podano nieistniejący identyfikator subskrypcji)

Adres usługi

- Platforma testowa:
 - Usługa pobierania danych i zarządzania jednostkami KAP:
<https://cutest.wrotapodlasia.pl/CU.WS.KUP/KupService.svc>
 - Usługa subskrypcji:
<https://cutest.wrotapodlasia.pl/CU.WS.KUP/SubscriptionService.svc>
- Platforma produkcyjna (tymczasowo – do czasu udostępnienia publicznego CU):
 - Usługa pobierania danych i zarządzania jednostkami KAP:
<https://cuprod.wrotapodlasia.pl/CU.WS.KUP/KupService.svc>
 - Usługa subskrypcji:
<https://cuprod.wrotapodlasia.pl/CU.WS.KUP/SubscriptionService.svc>
- Platforma produkcyjna (docelowo – od momentu udostępnienia publicznego CU):
 - Usługa pobierania danych i zarządzania jednostkami KAP:
<https://cu.wrotapodlasia.pl/CU.WS.KUP/KupService.svc>
 - Usługa subskrypcji:
<https://cu.wrotapodlasia.pl/CU.WS.KUP/SubscriptionService.svc>

Opis usługi ESP.WS

Usługa Cyfrowego Urzędu odpowiedzialna za przyjmowanie dokumentów składanych przez podmioty publiczne do Jednostek za potwierdzeniem przedłożenia.

Przekazanie dokumentu na skrzynkę podawczą jednostki

Metoda służąca do przekazania dokumentu na skrzynkę podawczą jednostki.

Nazwa metody: dorecz_jednostce

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
senderUnitId	Int	Tak	Identyfikator jednostki nadawcy
destinationUnitId	Int	Tak	Identyfikator jednostki odbiorcy
documentCU	DocumentCU	Tak	Dane dokumentu

Zwracany wynik: obiekt typu DeliveryResponse

Opis klas:

DocumentCU

Typ danych służący do przekazywania informacji o wniosku lub doręczeniu. Ponieważ jest wspólny dla kilku usług, to niektóre pola mają zastosowanie tylko w przypadku określonej metody.

Pola: nazwa	Typ	Czy wymagany?	Opis
-------------	-----	---------------	------

DocumentId	Nullable<int>	Nie	Identyfikator dokumentu
BinaryData	byte[]	Tak	Dane binarne dokumentu
Title	string	Tak	Tytuł dokumentu
DeliveryDeadline	Nullable<DateTime>	Nie	Termin doręczenia
ConnectedDocumentId	Nullable<int>	Nie	Identyfikator powiązanego dokumentu

DeliveryResponse

Typ danych służący do przekazywania informacji o wyniku operacji.

Pola: nazwa	Typ	Opis
Status	string	Status operacji (NEGATIVE/POSITIVE/ERROR)
DocumentId	int	Identyfikator nowo utworzonego dokumentu w CU
Message	string	Dodatkowe informacje, informacja diagnostyczna

Doręczenie dokumentu systemowi SEOD wybranej jednostki

Metoda będąca odpowiednikiem doręcz_jednostce dedykowana dla systemu SEOD. Przyjmuje ona identyfikatory SEOD zamiast identyfikatorów Jednostek.

Nazwa metody: dorecz_jednostce_seod

Adres usługi

- Platforma testowa: <https://cutest.wrotapodlasia.pl/CU.WS.ESP/Service.svc>
 - Platforma produkcyjna: <https://cuadmin.wrotapodlasia.pl/CU.WS.ESP/Service.svc>
- Adres jest dostępny w sieci wewnętrznej UMWP. Jeżeli nazwa domenowa jest nierozpoznawana należy dodać wpis w pliku hosts na komputerze, z którego wykonujemy połączenie:

```
10.200.3.30          cuadmin.wrotapodlasia.pl
```

Opis usługi DS.WS

Usługa Cyfrowego Urzędu odpowiedzialna za przyjmowanie doręczeń do Interesantów w trybie KPA, zwracanie potwierdzeń odbioru do systemów zewnętrznych oraz przyjmowanie dowolnych dokumentów do przekazania do Interesanta.

Doręczenie dokumentów do interesanta w trybie KPA

Metoda służąca do przekazania doręczenia do Interesanta w trybie KPA.

Nazwa metody: dorecz_osobie

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
documentCU	DocumentCU	Tak	Dane dokumentu
clientData	ClientData	Tak	Dane identyfikujące Interesanta
unitId	int	Tak	Identyfikator Jednostki nadawcy
updXml	string	Tak	Dane XML potwierdzenia odbioru (UPD) do podpisania przez Interesanta

Usługa przeprowadza weryfikację czy przekazane dane Interesanta są wystarczające, system zewnętrzny może podać je na jeden z poniższych sposobów:

- imię, nazwisko, PESEL i login OKI,
- adres email Interesanta.

Pobieranie UPD

Metoda służąca do pobrania przez system zewnętrzny potwierdzeń doręczeń, które zostały już podpisane przez Interesantów lub uznane za doręczone na podstawie prawa.

Nazwa metody: pobierz_UPD

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
unitId	int	Tak	Identyfikator Jednostki nadawcy

Zwracany wynik: lista obiektów typu **Upd**

Pobieranie UPD z potwierdzeniem doręczenia

Metoda służąca do pobrania przez system zewnętrzny potwierdzeń doręczeń, które zostały już podpisane przez Interesantów lub uznane za doręczone na podstawie prawa. Wywołanie tej metody nie oznacza automatycznie potwierdzeń jako przekazane do systemu zewnętrznego. W tym celu potwierdzenia poprawnego przyjęcia UPD, system zewnętrzny wywołuje metodę **potwierdz_UPD**.

Nazwa metody: pobierz_UPD_z_potwierdzeniem

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
unitId	int	Tak	Identyfikator Jednostki nadawcy

Zwracany wynik: lista obiektów typu **Upd**

Potwierdzenie pobrana UPD przez system zewnętrzny

Metoda służąca do potwierdzenia przez system zewnętrzny poprawnego przyjęcia UPD

Nazwa metody: potwierdz_UPD

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
updIds	List<int>	Tak	Identyfikatory przyjętych UPD

Zwracany wynik: bool – czy operacja wykonana prawidłowo.

Przekazanie dowolnego dokumentu na OKI

Metoda służąca do przekazania dowolnego dokumentu na OKI

Nazwa metody: przeslij_osobie

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
okiLogin	string	Tak	Login Interesanta w OKI
unitId	int	Tak	Identyfikator Jednostki nadawcy
documentCU	DocumentCU	Tak	Dane dokumentu

Zwracany wynik: **DeliveryResponse**

Doręczenie dokumentów do interesanta w trybie KPA (na potrzeby SEOD)

Odpowiednik metody **dorecz_osobie** dedykowany dla SEOD. Przyjmuje identyfikatory SEOD zamiast identyfikatorów Jednostek.

Nazwa metody: dorecz_osobie_seod

Przekazanie dowolnego dokumentu na OKI (na potrzeby SEOD)

Odpowiednik metody **prześlij_osobie** dedykowany dla SEOD. Przyjmuje identyfikatory SEOD zamiast identyfikatorów Jednostek.

Nazwa metody: przeslij_osobie_seod

Opis klas:

ClientData

Typ służący do przekazywania danych identyfikujących Interesanta.

Pola: nazwa	Typ	Czy wymagany?	Opis
FirstName	string	Nie	Imię Interesanta
LastName	string	Nie	Nazwisko Interesanta
PESEL	string	Nie	Numer PESEL Interesanta
Email	string	Nie	Adres Email Interesanta
OkiLogin	string	Nie	Login OKI Interesanta

Upd

Typ służący do przekazywania danych Urzędowego Potwierdzenia Doręczenia.

Pola: nazwa	Typ	Opis
Id	int	Identyfikator UPD w CU
CuDocumentId	int	Identyfikator doręczenia w CU
BinaryData	byte[]	Dane binarne UPD
IsSigned	bool	Czy doręczenie podpisane przez Interesanta
DeliveredByLaw	bool	Czy uznano za doręczone na podstawie prawa
CauseOfFailure	String	Przyczyna niepowodzenia (w przypadku niemożliwości doręczenia)

Adres usługi

- Platforma testowa: <https://cutest.wrotapodlasia.pl/CU.WS.DS/Service.svc>
- Platforma produkcyjna: <https://cuadmin.wrotapodlasia.pl/CU.WS.DS/Service.svc>

Adres jest dostępny w sieci wewnętrznej UMWP. Jeżeli nazwa domenowa jest nierozpoznawana należy dodać wpis w pliku hosts na komputerze, z którego wykonujemy połączenie:

10.200.3.30

cuadmin.wrotapodlasia.pl

Opis usługi CU.SeodIntegrator

Usługa odpowiadająca za integrację z SEOD

Aktualizacja danych sprawy

Metoda służąca do aktualizacji informacji o sprawie

Nazwa metody: UpdateCase

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
documentId	int	Tak	Identyfikator dokumentu
seodId	string	Tak	Identyfikator SEOD Jednostki
caseInformation	CaseInformation	Tak	Dane sprawy

Zwracany wynik: **CaseUpdateResponse**

Dodanie płatności

Metoda służąca do dodania płatności, którą mu wnieść interesant w związku z realizacją jego sprawy.

Nazwa metody: AddPayment

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
documentId	int	Tak	Identyfikator dokumentu
seodId	string	Tak	Identyfikator SEOD Jednostki
paymentInformation	PaymentInformation	Tak	Dane płatności

Zwracany wynik: **PaymentAddResponse**

Aktualizacja płatności

Metoda służąca do aktualizacji danych płatności

Nazwa metody: UpdatePayment

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
documentId	int	Tak	Identyfikator dokumentu
seodId	string	Tak	Identyfikator SEOD Jednostki
paymentInformation	PaymentInformation	Tak	Dane płatności

Dodanie użytkownika wewnętrznego do Cyfrowego Urzędu

Metoda służąca do dodania nowego użytkownika wewnętrznego do systemu Cyfrowy Urząd przez system SEOD.

Nazwa metody: AddUser

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
user	UserDto	Tak	Dane użytkownika

Zwracany wynik: **brak**

Aktualizacja użytkownika wewnętrznego Cyfrowego Urzędu

Metoda służąca do aktualizacji danych użytkownika wewnętrznego przez system SEOD.

Nazwa metody: UpdateUser

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
user	UserDto	Tak	Dane użytkownika. Należy pominąć pole Password, gdyż podczas aktualizacji nie jest brane pod uwagę.

Zwracany wynik: **brak**

Usunięcie użytkownika wewnętrznego Cyfrowego Urzędu

Metoda służąca do usuwania użytkowników wewnętrznych Cyfrowego Urzędu przez system SEOD. Usuwanie polega na dezaktywacji kont (konta nie są usuwane fizycznie z bazy danych).

Nazwa metody: DeleteUser

Parametry:

Nazwa parametru	Typ	Czy wymagany?	Opis
userName	string	Tak	Login użytkownika

Zwracany wynik: **brak**

Opis złożonych typów danych wykorzystanych w usłudze SeodIntegrator

CaseUpdateResponse

Typ danych służący do przekazywania informacji o wyniku operacji aktualizacji sprawy.

Pola: nazwa	Typ	Opis
Result	WsOperationResult (enumerator)	Wynik operacji (Negative = 0 / Positive = 1/ Error = 255)
Message	string	Dodatkowe informacje, informacja diagnostyczna

PaymentAddResponse

Typ danych służący do przekazywania informacji o wyniku operacji dodania nowej płatności.

Pola: nazwa	Typ	Opis
Result	WsOperationResult (enumerator)	Wynik operacji (Negative = 0 / Positive = 1/ Error = 255)
Message	string	Dodatkowe informacje, informacja diagnostyczna
PaymentId	Nullable<int>	Identyfikator nowej płatności

PaymentUpdateResponse

Typ danych służący do przekazywania informacji o wyniku operacji aktualizacji danych płatności.

Pola: nazwa	Typ	Opis
Result	WsOperationResult (enumerator)	Wynik operacji (Negative = 0 / Positive = 1/ Error = 255)
Message	string	Dodatkowe informacje, informacja diagnostyczna

UserDto

Typ danych opisujący użytkownika wewnętrznego

Pola: nazwa	Typ	Opis
UserName	string	Login użytkownika
Email	string	Email użytkownika
Firstname	string	Imię użytkownika
Lastname	string	Nazwisko użytkownika
SeodId	string	Identyfikator SEOD
Password	string	Hasło użytkownika (hash SHA512)

Adres usługi

- Platforma testowa:
<https://cutest.wrotapodlasia.pl/CU.WS.SeodIntegrator/Service.svc>
- Platforma produkcyjna:
<https://cuadmin.wrotapodlasia.pl/CU.WS.SeodIntegrator/Service.svc>