w ramach projektu –

Beneficjent:

|  |  |
| --- | --- |
| **Wersja:** | 2.03 |
| **Data wersji:** | 2023-03-22 |
| **Autor (rzy):** | Dominik Roszkowski |
| **Nazwa pliku:** | KREPTD\_API\_do\_udostępniania\_danych\_publicznych\_v203\_20230322.docx |
| **Akronim:** | KREPTD-DOC-API-PUB |

Tabela Metryka dokumentu

| Tytuł dokumentu: |  | | |
| --- | --- | --- | --- |
| Typ dokumentu: | Dokumentacja Specjalistyczna | | |
| Nazwa Projektu: |  | | |
| Właściciel dokumentu | Pentacomp Systemy Informatyczne | | |
| Lokalizacja dokumentu: | Lokalizacja SVN | | |
| Autor (rzy): |  | Numer wersji dokumentu: | 2.01 |
| Data utworzenia dokumentu: | 2018-08-13 |
| Sprawdził (a): | Krystian Bąk | Data sprawdzenia wersji: | 2023-03-22 |
| Zatwierdził (a) | Krystian Bąk | Data zatwierdzenia wersji: | 2023-03-22 |

Tabela Historia zmian dokumentu

| Nr wersji | Data wersji | Autor zmiany | Komentarz/Uwagi/Zakres zmian |
| --- | --- | --- | --- |
| 1.00 | 2018-08-13 | Dominik Roszkowski | Pierwsza wersja dokumentu |
| 2.00 | 2019-04-29 | Krystian Bąk | Aktualizacja metryki |
| 2.01 | 2021-12-20 | Krystian Bąk | Aktualizacja adresu produkcyjnego |
| 2.02 | 2022-08-31 | Krystian Bąk | Aktualizacja załączników do dokumentu |
| 2.03 | 2023-03-22 | Krystian Bąk | Aktualizacja załączników do dokumentu |

Tabela Dystrybucja dokumentu

| Nr wersji | Data dostarcz. | Adresat | Funkcja/Stanowisko |
| --- | --- | --- | --- |
| 1.00 | 2018-10-01 | Monika Banaszek | Kierownik Projektu KREPTD |
| 2.00 | 2019-04-29 |  | Kierownik Projektu KREPTD |
| 2.01 | 2021-12-20 | Rafał Godlewski | Kierownik Projektu KREPTD |
| 2.02 | 2022-08-31 | Rafał Godlewski | Kierownik Projektu KREPTD |
| 2.03 | 2023-03-22 | Rafał Godlewski | Kierownik Projektu KREPTD |

Tabela Akceptacja dokumentu

| Nr wersji | Data akceptacji | Akceptujący | Zakres | Podpis | Firma |
| --- | --- | --- | --- | --- | --- |
| 2.03 | 2023-03-22 | Krystian Bąk |  |  | Pentacomp Systemy Informatyczne |
| 2.03 | 2023-03-22 | Rafał Godlewski |  |  |  |

Spis treści

[1. Wstęp 7](#_Toc132096347)

[1.1. Cel dokumentu 7](#_Toc132096348)

[1.2. Odbiorcy produktu 7](#_Toc132096349)

[1.3. Załączniki 7](#_Toc132096350)

[2. Przeznaczenie API 9](#_Toc132096351)

[3. Architektura i opis działania API 10](#_Toc132096352)

[3.1. Uwierzytelnienie i autoryzacja 10](#_Toc132096353)

[3.1.1. Wnioskowanie o certyfikat 11](#_Toc132096354)

[3.1.2. Dostęp do danych 12](#_Toc132096355)

[3.2. Słowniki 12](#_Toc132096356)

[3.3. Obsługa błędów 13](#_Toc132096357)

[3.4. Przykładowa weryfikacja dostępu do usługi 13](#_Toc132096358)

[3.5. Przykładowe wywołania usługi 17](#_Toc132096359)

[3.6. Pliki kontraktu 19](#_Toc132096360)

[3.7. Przykładowy projekt SOAP UI 19](#_Toc132096361)

[4. Spis tabel 20](#_Toc132096362)

[5. Spis rysunków 21](#_Toc132096363)

# Wstęp

## Cel dokumentu

Celem niniejszego dokumentu jest opis działania usługi sieciowej (API) do udostępniania danych publicznych klientom implementującym bezpośrednią komunikację z systemem KREPTD.

## Odbiorcy produktu

Odbiorcami niniejszego produktu są zespoły projektowe, zamierzające implementować komunikację z systemem KREPTD.

## Załączniki

Dokument posiada następujące załączniki:

* CitizenApi.wsdl – kontrakt usługi sieciowej w formacie WSDL (wersja 5.0),
* CitizenApi.xsd – opis struktur wykorzystywanych w usłudze sieciowej w formacie XSD (wersja 5.0).
* CITIZEN-API-soapui-project.xml – przykładowy projekt i wywołania usług w oprogramowaniu SOAP UI
* CITIZEN-API.html – dokumentacja dostępnych operacji w usłudze
* Dane testowe

Tabela Słownik pojęć

| Termin | Wyjaśnienie |
| --- | --- |
| API | (ang. Application Programming Interface) Sposób, rozumiany jako ściśle określony zestaw reguł i opisów w jaki programy komputerowe komunikują się między sobą. W kontekście systemu KREPTD – usługa sieciowa dla systemów zewnętrznych. |
| GITD | Główny Inspektorat Transportu Drogowego |
| Klient API | System informatyczny integrujący się z usługą API |
| KPK | Krajowy Punkt Kontaktowy w ramach GITD |
| KREPTD, Rejestr, System | Krajowy Rejestr Elektroniczny Przedsiębiorców Transportu Drogowego |
| WebService | Usługa sieciowa |
| WSDL | (ang. Web Services Description Language), język do definiowania usług internetowych oparty na XML, także plik definicji (kontraktu). |
| XML | Uniwersalny język znaczników przeznaczony do reprezentowania różnych danych w strukturalizowany sposób (ang. Extensible Markup Language). |
| XSD | Plik w formacie XML, opisujący dedykowaną strukturę plików XML specyficznych dla danego systemu informatycznego. |
| Zamawiający | Główny Inspektorat Transportu Drogowego (GITD) |

# Przeznaczenie API

System KREPTD udostępnia usługę sieciową (WebService), pozwalającą na implementację klienta, który może wykorzystywać funkcjonalności strony www dla obywateli bez pośrednictwa interfejsu graficznego aplikacji. Pozwala to zewnętrznym instytucjom zintegrować własne systemy i aplikacje z systemem KREPTD.

# Architektura i opis działania API

API opisane jest plikiem WSDL (**CitizenApi.wsdl**), zawierającym dostępne operacje oraz ich parametry, oraz towarzyszącym plikiem XSD (**CitizenApi.xsd**), posiadającym definicje wykorzystywanych struktur danych (np. model zarządzającego, przedsiębiorcy itp.).

Implementowane standardy:

* SOAP 1.1,
* WS-I Basic Profile 1.1,
  + Zgodność z tym standardem maksymalizuje liczbę potencjalnych technologii, które mogą skomunikować się z usługą.

Adres usługi testowej:

* [https://](https://api.gitd.gov.pl/KreptdApi.svc)91.216.125.189:8084/CitizenApi.svc

Adres usługi produkcyjnej:

* [https://obywatele.kreptd.gitd.gov.pl:8447/](https://obywatele.kreptd.gitd.gov.pl:8447/" \t "_blank" \o "https://obywatele.kreptd.gitd.gov.pl:8447/)CitizenApi.svc

Całość interfejsu, nazwy operacji, modele i opisy są w języku polskim, co powinno znacząco ułatwić potencjalną integrację. Wszystkie operacje są synchroniczne i dane są aktualne na moment pobrania, jeśli tylko proces uwierzytelnienia i autoryzacji przebiegnie poprawnie. Możliwe jest wyłącznie pobieranie i wyszukiwanie danych. Dodatkowo, liczba rekordów zwracanych przez operacje wyszukiwania ograniczana jest parametrem systemowym.

Usługa NIE wymaga konkretnych prefixów przestrzeni nazw w przesłanych komunikatach XML.

## Uwierzytelnienie i autoryzacja

W celu poprawnego wywołania operacji API, klient usługi musi poprawnie przejść pełny proces uwierzytelnienia i autoryzacji. Dostęp do usługi wymaga SSL oraz wykorzystania certyfikatu klienckiego X.509, wystawionego przez zaufane centrum certyfikacji. Certyfikat ten musi zostać zarejestrowany w systemie KREPTD jako certyfikat dostępu dla danego klienta API publicznego. Każdy klient API publicznego, zdefiniowany w Systemie za pomocą dedykowanej funkcjonalności musi posiadać unikalny certyfikat. Jeśli weryfikacja certyfikatu przebiegnie poprawnie, weryfikowane jest czy klient API posiada aktywny dostęp.

Błąd uwierzytelnienia lub autoryzacji, na jakimkolwiek etapie przedstawionego procesu, skutkuje zwróceniem do klienta kodu HTTP 403, w szczególności brak certyfikatu klienckiego będzie skutkował kodem HTTP 403.7.

Główny Inspektorat Transportu Drogowego wydaje przedmiotowe certyfikaty oraz nadaje odpowiednie uprawnienia systemom informatycznym, wnioskującym o integrację z systemem KREPTD.

### Wnioskowanie o certyfikat

Do autoryzacji klienta API nalezy posiadać plik (.pfx) zawierający certyfikat i klucz prywatny. Plik ten generowany jest na podstawie certyfikatu publicznego (.cer, .crt). W celu zawnioskowania o Certyfikat publiczny należy wysłać żądanie podpisania certyfikatu (CSR) do GITD. W tym celu należy wygenerować plik CSR (.csr). Do wygenerowania zarówno pliku CSR jak i klucza prywatnego można przykładowo użyć oprogramowania OpenSSL.

Przykładowy proces generowania certyfikatu przy użyciu OpenSSL:

* Otwórz wiersz poleceń
* Wygeneruj plik CSR poprzez użycie komendy:
* *openssl req -new -newkey rsa:2048 -nodes -keyout nasza\_organizacja.key -out nasza*\_organizacja.csr
* Po wykonaniu powyższej komendy należy podać danych do CSR. Przykład poniżej:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek Generowanie certyfikatu - przykładowe dane

* Parametry:
  + Country Name – podaj dwuliterowy kod kraju np. „PL”.
  + State or Province Name – należy podać nazwę województwa np. „Mazowieckie”
  + Locality Name – należy podać nazwę miejscowości, w której mieści się siedziba firmy np. „Warszawa”
  + Organization Name (O) – należy podać pełną nazwę firmy (wskazaną we wniosku)
  + Organizational Unit Name – pozostaw to pole puste
  + Common Name – należy wpisać nazwę systemu, dla którego wnioskujesz o dostęp, wartość powinna być uzależniona od środowiska np.

integracyjnego: KREPTD-NP.

produkcyjnego: KREPTD

* + Email Address – pozostaw to pole puste.
* Wynikiem powyższych otrzymujesz dwa pliki:
  + \*.csr
  + \*.key

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek Generowanie certyfikatu - pliki

* Do GITD należy wysłać tylko plik z rozszerzeniem .csr (nasza\_organizacja.csr). W odpowiedzi od GITD otrzymasz plik certyfikatu (.cer lub.crt)
* Z otrzymanego pliku certyfikatu generujemy plik z kluczep prywatnym (.pfx), aby to zrobić wykonaj poniższą komendę. Parametry powinny wskazywać na otrzymany plik .cer lub .crt i wcześniej wygenerowany plik .key. Zostaniemy zapytani o podanie hasła, które zabezpieczy nasz plik pfx.
* *openssl pkcs12 -export -out nasza\_organizacja.pfx -inkey nasza\_organizacja.key -in nasza\_organizacja.cer*

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek Generowanie certyfikatu - generowanie pfx

* W wyniku powyższych działań otrzymujesz plik pfx którym będziesz autoryzować się w KREPTD API

### Dostęp do danych

Wywołanie każdej operacji rejestrowane jest w systemie KREPTD. Dostęp do danych ograniczony jest w sposób analogiczny jak w przypadku aplikacji www dla obywateli tj. dostępna jest tylko aktualna wersja danych bez uwzględnienia wybranych danych wrażliwych.

## Słowniki

Wartości słownikowe obecne w parametrach wyszukiwania oraz w zwracanych danych są zdefiniowane w schemacie XSD. Uzasadnione jest to niewielką liczbą użytych słowników i stosunkowo dużą niezmiennością użytych wartości.

## Obsługa błędów

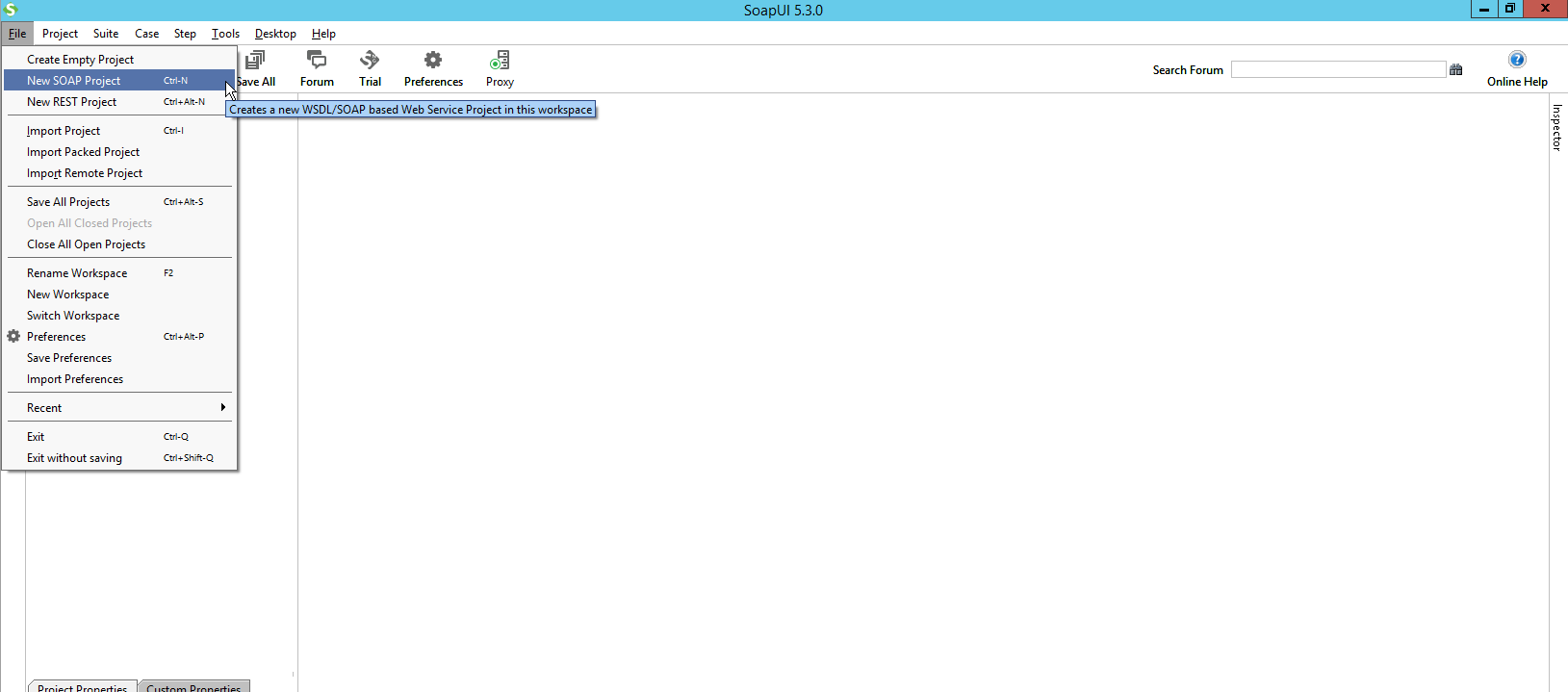
Nie licząc błędów dotyczących uwierzytelnienia i autoryzacji, API zawsze informuje o błędach za pomocą komunikatu SOAP Fault (z dedykowaną zawartością) i kodem HTTP 500. Wszystkie komunikaty błędów są w języku polskim oraz zawierają kod statusu błędu (zdefiniowany   
w WSDL), który pozwala na kategoryzację i lepszą diagnostykę błędów po stronie zarówno klienta jak i serwera.

Zanim żądanie zostanie poprawnie obsłużone, przesłany komunikat XML SOAP jest weryfikowany z załączonym do dokumentu schematem XSD.

## Przykładowa weryfikacja dostępu do usługi

W celu weryfikacji dostępu do usługi należy wykonać poniższe kroki przedstawione na zrzutach ekranu (przykład jest oparty na wykorzystaniu narzędzia SoapUI) :

* Krok pierwszy



Rysunek 4 SoapUI - New Soap Project

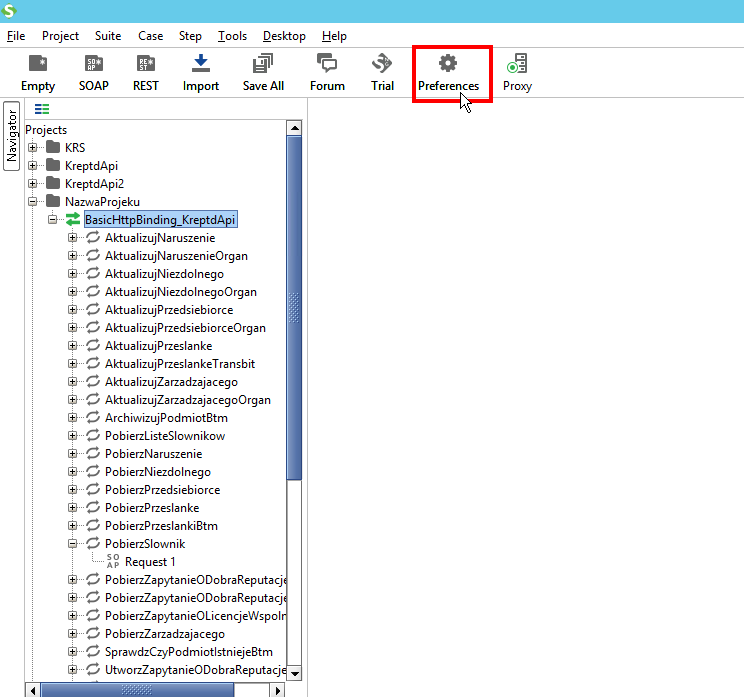
* Krok drugi

W polu Initail WSDL powinien zostać podany adres usługi integracyjnej lub produkcyjnej.

Obraz zawierający tekst

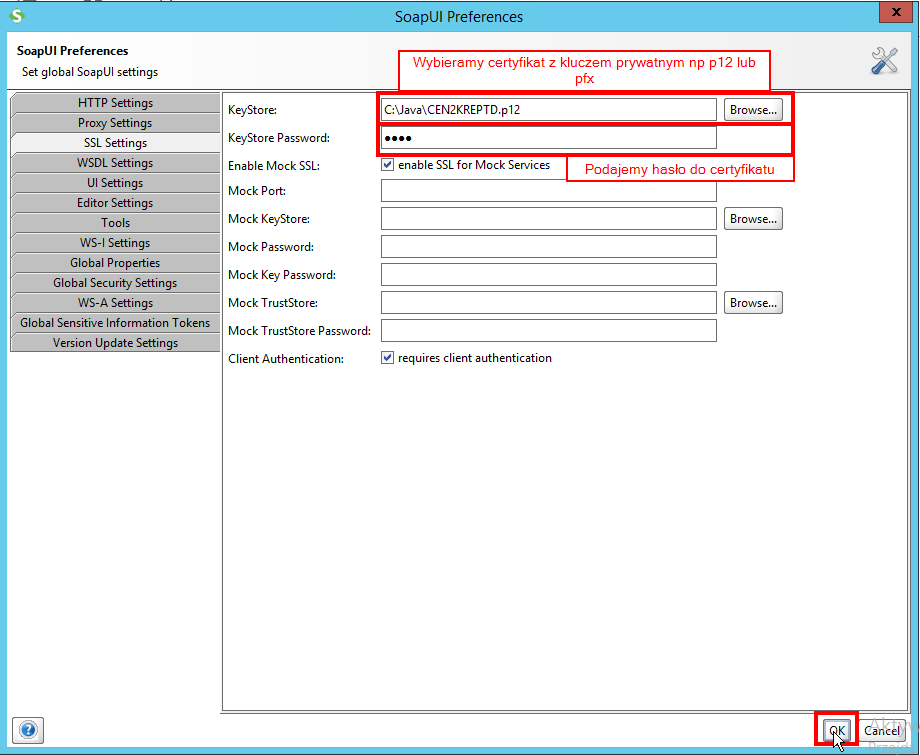
Opis wygenerowany automatycznie

Rysunek SoapUI - Project Name

* Krok trzeci

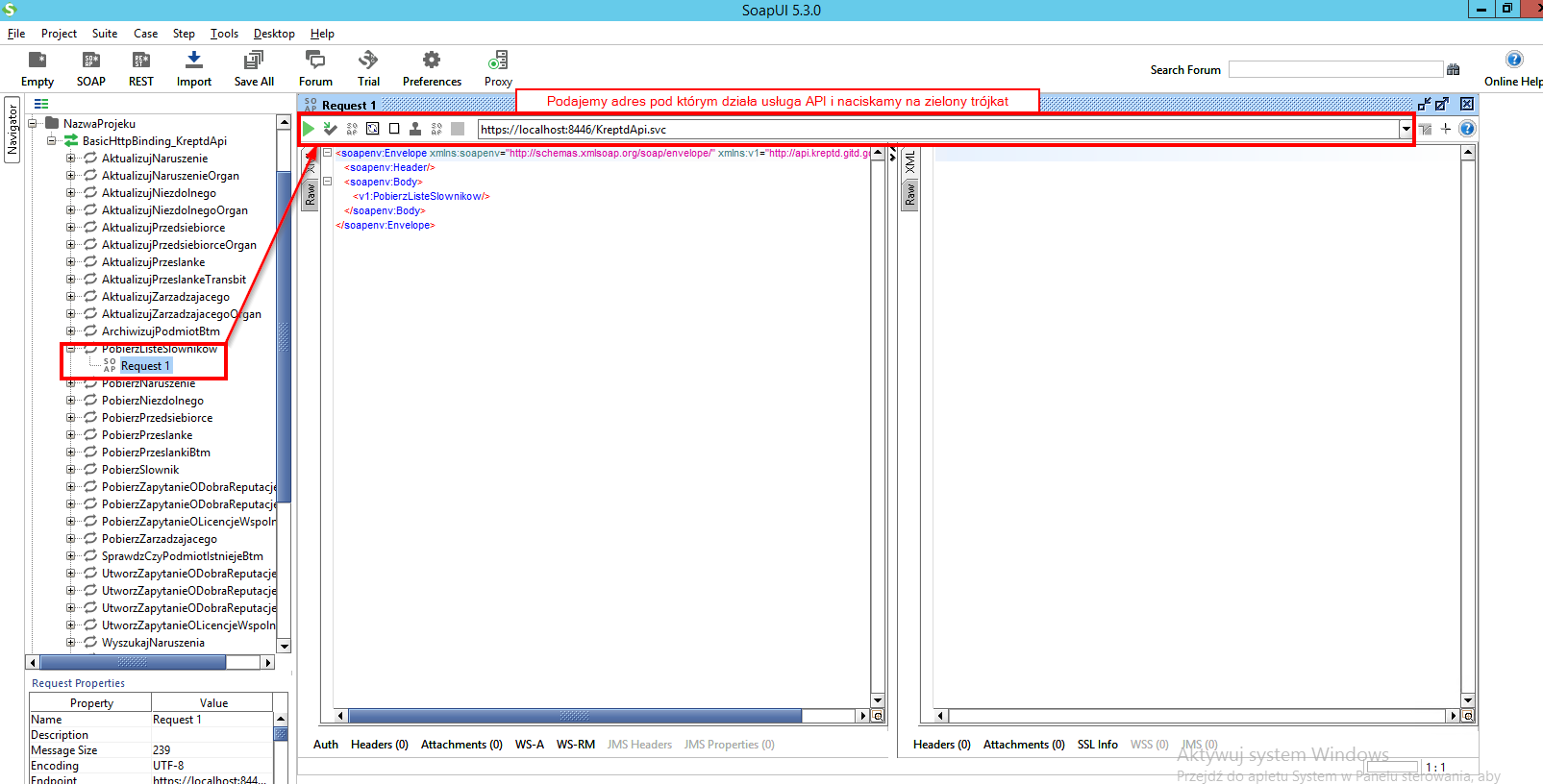
Rysunek SoapUI - Preferences

* Krok czwarty



Rysunek SoapUI - SSL Settings

* Krok piąty



Rysunek SoapUI - Reqest

* Krok szósty

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek SoapUI - Response

## Przykładowe wywołania usługi

**Przykład 1: Walidacja ze schematem nie przebiegła pomyślnie.**

Odpowiedź serwera:

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

<s:Body>

<s:Fault>

<faultcode>s:Client</faultcode>

<faultstring xml:lang="pl-PL">Żądanie zakończyło się błędem</faultstring>

<detail>

<SzczegolyBledu xmlns="http://api.citizen.kreptd.gitd.gov.pl/v1.0" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

<KodBledu>StrukturaKomunikatu</KodBledu>

<KomunikatBledu>Struktura XML komunikatu SOAP jest niepoprawna</KomunikatBledu>

<ListaBledowStruktury>

<BladStruktury> <Sciezka>**/Envelope/Body/WyszukajPrzedsiebiorce/parametryWyszukiwaniaPrzedsiebiorcow/NumerReferencyjny**</Sciezka>

<Opis>**Element 'http://api.citizen.kreptd.gitd.gov.pl/v1.0/types:NumerReferencyjny' jest nieprawidłowy. Wartość 'P/2018/0300022' jest nieprawidłowa przy uwzględnieniu jego typu danych** **http://api.citizen.kreptd.gitd.gov.pl/v1.0/types:NumerReferencyjnyPrzedsiebiorcyTyp' - Błąd ograniczenia elementu Pattern..</**Opis>

</BladStruktury>

</ListaBledowStruktury>

</SzczegolyBledu>

</detail>

</s:Fault>

</s:Body>

</s:Envelope>

Elementy <faultcode> i <faultstring> zawsze mają taką samą zawartość. Szczegóły dedykowane systemowi KREPTD znajdują się w elemencie <SzczegolyBledu>. Kod błędu jest jednym z możliwych kodów podanych w pliku WSDL, zaś komunikat błędu opisuje błąd w żądaniu. W przypadku błędów walidacji schematu XSD, zwracana jest dodatkowo <ListaBledowStruktury>.

**Przykład 2: Walidacja ze schematem XSD przebiegła poprawnie, lecz wyszukiwanie nie dało żadnych rezultatów.**

Odpowiedź serwera:

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

<s:Body>

<WyszukajPrzedsiebiorceResponse xmlns="http://api.citizen.kreptd.gitd.gov.pl/v1.0">

<listaPrzedsiebiorcowDaneWyszukane xmlns:a="http://api.citizen.kreptd.gitd.gov.pl/v1.0/types" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"/>

</WyszukajPrzedsiebiorceResponse>

</s:Body>

</s:Envelope>

**Przykład 3: Walidacja ze schematem XSD przebiegła poprawnie, lecz przedsiębiorca o podanym numerze referencyjnym nie znajduje się w ewidencji (argument niepoprawny biznesowo).**

Odpowiedź serwera:

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

<s:Body>

<s:Fault>

<faultcode>s:Client</faultcode>

<faultstring xml:lang="pl-PL">Żądanie zakończyło się błędem</faultstring>

<detail>

<SzczegolyBledu xmlns="http://api.citizen.kreptd.gitd.gov.pl/v1.0" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

<KodBledu>**BrakDanych**</KodBledu>

<KomunikatBledu>**Brak przedsiębiorcy o numerze referencyjnym P/2018/03000221**</KomunikatBledu>

<ListaBledowStruktury i:nil="true"/>

</SzczegolyBledu>

</detail>

</s:Fault>

</s:Body>

</s:Envelope>

## Pliki kontraktu

Oba pliki kontraktu (WSDL: CitizenApi.wsdl i XSD: CitizenApi.xsd) są załącznikiem do niniejszego dokumentu projektu.

## Przykładowy projekt SOAP UI

Załącznikiem do dokument jest przykładowy projekt SOAP UI zawierający wygenerowane na podstawie WSDL dostępne w usłudze operacje. Zawiera także przykładowe ich wywołania. Do działania projektu należy w konfiguracji programu SOAP UI ustawić otrzymany certyfikat autoryzacyjny wydany przez Główny Inspektorat Transportu Drogowego. W tym celu należy otworzyć kolejno File -> Preferences -> SSL Settings -> i w opcje KeyStore wybrać plik pfx, a także w opcji KeyStore Password podać hasło co certyfikatu.

# Spis tabel

[Tabela 1 Metryka dokumentu 3](#_Toc132096364)

[Tabela 2 Historia zmian dokumentu 3](#_Toc132096365)

[Tabela 3 Dystrybucja dokumentu 3](#_Toc132096366)

[Tabela 4 Akceptacja dokumentu 3](#_Toc132096367)

[Tabela 5 Słownik pojęć 7](#_Toc132096368)

# Spis rysunków

[Rysunek 1 Generowanie certyfikatu - przykładowe dane 11](#_Toc132096369)

[Rysunek 2 Generowanie certyfikatu - pliki 12](#_Toc132096370)

[Rysunek 3 Generowanie certyfikatu - generowanie pfx 12](#_Toc132096371)

[Rysunek 4 SoapUI - New Soap Project 13](#_Toc132096372)

[Rysunek 5 SoapUI - Project Name 14](#_Toc132096373)

[Rysunek 6 SoapUI - Preferences 14](#_Toc132096374)

[Rysunek 7 SoapUI - SSL Settings 15](#_Toc132096375)

[Rysunek 8 SoapUI - Reqest 16](#_Toc132096376)

[Rysunek 9 SoapUI - Response 16](#_Toc132096377)