Nemzeti Élelmiszerlánc-biztonsági Hivatal

API csatlakozás útmutató

v2.0

Dokumentumtörténet

Verziószám	Dátum	Változások
1.0	2023.05.04.	Alapváltozat
2.0	2023.09.01.	Token adatok megadása fejezet kiegészítés

Tartalomjegyzék

1 Bevezetés

A dokumentum célja, hogy segítséget nyújtson az API csatlakozáshoz, valamint a lehetséges hiányosságok feltárásához.

2 Kommunikáció típusa

Az API SOAP 1.1-es protokollt használ, az alábbi linken található leírás a protokollról:

https://www.w3.org/TR/2000/NOTE-SOAP-20000508/

https://en.wikipedia.org/wiki/SOAP

Az elsődleges tesztekhez érdemes a SoapUI használata, amely letölthető az alábbi linkről:

https://www.soapui.org/downloads/soapui/

3 Csatlakozás módja

Kétféle csatlakozási lehetőség van az API-hoz:

- IP cím szűréssel
- Tanúsítvány alapú hitelesítéssel

3.1 IP cím szűrés

Amennyiben a kapcsolódni kívánó fél garantálni tudja, hogy állandó IP cím(ek)ről fogják hívni az API végpontokat, elég a Nébih számára leadni az IP címe(ke)t.
Ebben az esetben nincs más teendő, a címek felvételre kerülnek az engedélyezettek közé és elérhetőek az egyes végpontok.

Elérések:

- Szolgáltatások:
    - Teszt: https://intesztfelirapi.nebih.gov.hu/peif/aap/services/{serviceName}
    - Éles: https://felirapi.nebih.gov.hu/peif/aap/services/{serviceName}
- WSDL
    - Teszt: https://intesztfelirapi.nebih.gov.hu/peif/aap/services/{serviceName}?wsdl
    - Éles: https://felirapi.nebih.gov.hu/peif/aap/services/{serviceName}?wsdl

Amennyiben a csatlakozás után a szolgáltatás wsdl címét meghívva letöltődik a wsdl állomány, a csatlakozás megfelelő.

3.2 *Tanúsítvány alapú hitelesítés*

A tanúsítvány alapú hitelesítés során a Nébih állít ki egy-egy X.509 tanúsítványt és adja át a crt/key fájlokat a fejlesztőknek, az X.509-ről információ az alábbi linken található:

https://en.wikipedia.org/wiki/X.509

Ezután a fejlesztőknek gondoskodni kell róla, hogy a szoftverbe megfelelően beépítve, a tanúsítvánnyal hívják meg a szolgáltatásokat.

Elérések:

- Szolgáltatások:
  - Teszt: https://intesztfelircertapi.nebih.gov.hu/peif/aap/services/{serviceName}
  - Éles: https://felircertapi.nebih.gov.hu/peif/aap/services/{serviceName}
- WSDL
  - Teszt: https://intesztfelircertapi.nebih.gov.hu/peif/aap/services/{serviceName}?wsdl
  - Éles: https://felircertapi.nebih.gov.hu/peif/aap/services/{serviceName}?wsdl

Amennyiben a csatlakozás után a szolgáltatás wsdl címét meghívva letöltődik a wsdl állomány, a csatlakozás megfelelő.

Ha a csatlakozás során a tanúsítvány nem érvényes, úgy HTTP válasz nélkül SSL hibával megszakad a kapcsolat. SoapUI (és más Java alapú kliensek) esetében a hiba üzenetek:

- Visszavont tanúsítvány esetén:

```
javax.net.ssl.SSLHandshakeException: Received fatal alert: certificate_revoked
```

- Rossz tanúsítvány esetén:

```
javax.net.ssl.SSLHandshakeException: Received fatal alert: certificate_required
```

Tekintettel arra, hogy a tanúsítvány alapú csatlakozás nem a leggyakoribb, így bemutatunk két módot. Az elsőben SoapUI eszköz paraméterezése látható, a másodikban egy egyszerű Java nyelven írt példa kód található.

3.2.1 SoapUI példa

A SoapUI segítségével történő teszteléshez szükség van egy PKCS#12 formátumú tanúsítvány fájlra, mely létrehozása az alábbi parancs használható:

```
openssl pkcs12 -export -in tanusitvany.crt -inkey kulcs.key -out tanusitvany.p12 -name tanusitvany -password pass:jelszo123
```

Az elkészült tanúsítványt hozzá kell adni a SoapUI projekthez, mely lépései a következők:

1. Nyisd meg a Project View nézetet
2. Navigálj a WS-Security Configurations/Keystores fülre
3. Add hozzá az elkészült .p12 kiterjesztésű fájlt a hitelesítési jelszó beírásával

Ezután a tanúsítvánnyal hitelesíteni kívánt kérés tulajdonságai között található SSL Keystore mezőnél kiválaszthatóvá válik a projekthez hozzáadott tanúsítvány.

3.2.2 Java példa

Egy java alkalmazásba történő beépítés esetén szükség van egy PKCS#12 formátumú tanúsítvány fájlra, mely létrehozása az alábbi parancs használható:

```
openssl pkcs12 -export -in tanusitvany.crt -inkey kulcs.key -out tanusitvany.p12 -name tanusitvany -password pass:jelszo123
```

Az alkalmazásban a következő függőségekre van szükség:

- org.apache.cxf:cxf-core
- org.apache.cxf:cxf-rt-frontend-jaxws
- org.apache.cxf:cxf-rt-transports-http
- org.apache.cxf:cxf-rt-ws-security

A tesztelés során az Apache CXF keretrendszer 3.5.5 számú verziója történt felhasználásra.
A megoldás előfeltétele a wsdl fájlból készített java kliens, mert tartalmazza a port osztályt. Az alábbi kódrészlet az ebből a port objektumból készített példányt paraméterezi fel megfelelően.

```
Client client = ClientProxy.getClient(port);
HTTPConduit httpConduit = (HTTPConduit) client.getConduit();
TLSClientParameters tlsCP = new TLSClientParameters();

KeyStore.Builder builder = KeyStore.Builder.newInstance("PKCS12", null, new File(P12_FILE), new KeyStore.PasswordProtection(CERT_PW.toCharArray()));
KeyManagerFactory fac = KeyManagerFactory.getInstance(KeyManagerFactory.getDefaultAlgorithm());
fac.init(builder.getKeyStore(), CERT_PW.toCharArray());
tlsCP.setKeyManagers(fac.getKeyManagers());
httpConduit.setTlsClientParameters(tlsCP);

BindingProvider bindingProvider = (BindingProvider) port;
Map<String, Object> requestContext = bindingProvider.getRequestContext();
requestContext.put("ws-security.username", USERNAME);
requestContext.put("ws-security.password", PASSWORD);
```

Az előző kódrészletben a következő szöveges változók definiálása szükséges:

- P12_FILE: a PKCS#12 fájl elérési útvonala
- CERT_PW: a PKCS#12 fájl hitelesítési jelszava
- USERNAME: a szolgáltatás használatához szükséges felhasználónév
- PASSWORD: a szolgáltatás használatához szükséges jelszó

A kódrészlet az alábbi import-okat használja:

```
import java.io.File;
import java.security.KeyStore;
import java.util.Map;
import javax.net.ssl.KeyManager;
import javax.net.ssl.KeyManagerFactory;
import javax.xml.ws.BindingProvider;
import org.apache.cxf.configuration.jsse.TLSClientParameters;
import org.apache.cxf.endpoint.Client;
import org.apache.cxf.frontend.ClientProxy;
import org.apache.cxf.transport.http.HTTPConduit;
```

Az Apache CXF keretrendszer WS-SecurityPolicy moduljának alapértelmezett beállítása mustUnderstand="1" érték. A paraméter értékének hamis értékre történő módosítására az alábbi interceptor osztály használható:

```
import java.util.List;
import org.apache.cxf.binding.soap.SoapHeader;
import org.apache.cxf.binding.soap.SoapMessage;
import org.apache.cxf.binding.soap.interceptor.AbstractSoapInterceptor;
import org.apache.cxf.headers.Header;
import org.apache.cxf.interceptor.Fault;
import org.apache.cxf.phase.Phase;


public class AddressingMustUnderstandInterceptor extends
AbstractSoapInterceptor {
   public AddressingMustUnderstandInterceptor () {
         super(Phase.WRITE);
   }


   @Override
   public void handleMessage (SoapMessage message) throws Fault {
         List<Header> list = message.getHeaders();
         list.stream().forEach(header ->
((SoapHeader)header).setMustUnderstand(false));
   }
}
```

Az interceptor osztályt az alábbi módon lehet beállítani a port objektumon keresztül elküldött kérések módosítására:

```
Client client = ClientProxy.getClient(port);
client.getOutInterceptors().add(new AddressingMustUnderstandInterceptor());
```

A bemutatott kódrészletek alkalmazásával a port osztályon keresztül elküldhetőek a kérések a megszokott módon, immár tanúsítvánnyal hitelesítve.

4 Autentikáció és autorizáció

A szolgáltatások használatához szükséges rendelkezni felhasználónév/jelszó párossal és az egyes szolgáltatásokhoz tartozó jogosultsággal.

A felhasználói adatokat a Web Services Security (WSS) alapján a UsernameToken elemben szükséges szerepeltetni, a WSS-ről és a UsernameToken-ről a következő linkeken található részletes leírás:

http://docs.oasis-open.org/wss-m/wss/v1.1.1/os/wss-SOAPMessageSecurity-v1.1.1-os.html

https://www.oasis-open.org/committees/download.php/13392/wss-v1.1-spec-pr-UsernameTokenProfile-01.htm

Az alábbi példában látható, hogy miként kell szerepelnie az adatoknak a SOAP Header-ben:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    <soapenv:Header>
      <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wsse:UsernameToken wsu:Id="UsernameToken-ECF651CEBDD642A5A2168054549918916">
          <wsse:Username>felhasználónév</wsse:Username>
          <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">jelszó</wsse:Password>
        </wsse:UsernameToken>
      </wsse:Security>
    </soapenv:Header>
    <soapenv:Body>
        ....
    </soapenv:Body>
</soapenv:Envelope>
```

A hívások során először ellenőrzi az API, hogy a kapott felhasználónév és jelszó páros megfelelő-e, ezután pedig ellenőrzi, hogy a felhasználónak van-e jogosultsága az adott szolgáltatás használatához.

A felhasználót és hozzá a jogosultságokat a Nébih által kért formátumban szükséges megigényelni.

5 Token adatok megadása

Néhány kiajánlott szolgáltatás esetében az API a megadott autentikációs adatokon kívül ellenőriz egy tokent is. Ez a token számokból és betűkből álló egyedi kódsor, amely az adatszolgáltatást beküldő személyhez tartozik és az egyedi beazonosítását garantálja.,

A token igénylését az Ügyfélprofil Rendszer ügykatalógusában, az API szolgáltatást nyújtó szakterület alatt található „Token igénylése külső szoftverből történő adatbeküldéshez” ügy elérésével lehet megtenni.

A token érvényességi ideje az igényléstől számított 1 év.

A tokennek a beérkező üzenet „callContext” elemének a „token” adatában szükséges szerepelnie az alábbi módon:

<callContext>

.

.

<token>TOKEN_HELYE<token>

.

</callContext>

Amennyiben olyan token adat érkezik, amely nem létezik vagy lejárt, úgy az API megtagadja a kiszolgálást, ilyenkor 401-es HTTP státusz kóddal és „Unathorized” üzenettel tér vissza.

6 Lehetséges válaszüzenetek

Az API biztonsági okokból nem ad ki információt az autentikációból fakadó problémákról.

- HTTP status 404-es hibaüzenetet ad amennyiben a felhasználónév/jelszó páros hibás, illetve amennyiben IP cím szűréssel csatlakoznak és olyan IP címről hívják a szolgáltatást, amely nem szerepel az engedélyezettek között.
- HTTP status 401-es hibaüzenetet ad amennyiben a csatlakozás sikeres, a felhasználónév/jelszó páros létezik azonban az adott szolgáltatáshoz nincs jogosultsága a felhasználónak. Ebben az esetben az alábbi SOAP Fault üzenetet adja vissza a rendszer az előzőekben írt státusz kóddal:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <soap:Fault>
         <faultcode xmlns:ns1="http://ws.apache.org/wss4j">ns1:Security Error</faultcode>
         <faultstring>Unauthorized</faultstring>
      </soap:Fault>
   </soap:Body>
</soap:Envelope>
```

- Megfelelő kérés esetén HTTP 200-as státusszal tér vissza az API és a response body-ban a válasz üzenet szerepel.
- Nem megfelelő tanúsítvány esetén nincs HTTP válasz (lásd 3.2fejezetben)

7 Probléma felderítése

Amennyiben a csatlakozás során probléma lép fel először az alábbi pontokat szükséges ellenőrizni:

- Csatlakozás módja:
    - IP cím esetén a leadott és beállított címről történik a szolgáltatás hívása.
    - Tanúsítvány esetén a megfelelő tanúsítvány van használatban a szolgáltatás hívása során.
- Az elküldött üzenet header elemében megfelelően szerepelnek a felhasználói adatok.
- A felhasználó rendelkezik a szolgáltatáshoz szükséges jogosultsággal.
- Az elküldött üzenet megfelel a wsdl-ben szereplő xsd-nek

Amennyiben a probléma továbbra is fennáll, a hibakereséshez az alábbi adatokat szükséges megküldeni a Nébih részére (api@nebih.gov.hu):

- Szolgáltatás címe
- Felhasználónév, amivel a szolgáltatás meghívásra került
- IP cím, ahonnan a szolgáltatás meghívásra került
- A szolgáltatás hívásának pontos ideje
- A teljes request xml állomány
- A válasz (hibakód és/vagy a response xml)