**Smoke szolgáltatás**

**használata**

Készült az ASP2 Adattárház projekthez

**Dokumentum történet**

| Verzió | Dátum | Státusz | A MEGVÁLTOZOTT RÉSZEK, a módosítás leírása | Készítő, Módosító |
| --- | --- | --- | --- | --- |
| 1.0 | 2018.04.03 |  | Első verzió | Kiss Attila Harri |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Műszaki ellenőrzést végezték**

| Dátum | Ellenőrizte | Cég | Aláírás |
| --- | --- | --- | --- |
|  |  | T-Systems Magyarország Zrt |  |
|  |  |  |  |

**Jóváhagyók**

|  | T-Systems részéről | Ügyfél részéről |
| --- | --- | --- |
| Név |  |  |
| Beosztás |  |  |
| Aláírás |  |  |
| Dátum |  |  |

**Dokumentum adatvédelmi besorolása**

|  |  |
| --- | --- |
| Minősítés |  |
| Minősítést végezte |  |
| Minősítés érvényessége |  |
| Betekintő |  |

Tartalom

[1. Bevezetés 4](#_Toc510704195)

[1.1 Jelen dokumentum célja 4](#_Toc510704196)

[2. Tesztelő eszköz 4](#_Toc510704197)

[2.1 SoapUI 4](#_Toc510704198)

[3. Smoke tesztelés előtt elvégzendő feladatok 6](#_Toc510704199)

[3.1 Projekt importálás 6](#_Toc510704200)

[3.2 Szolgáltatás címének megadása 7](#_Toc510704201)

[3.3 Minta kérések futtatása 9](#_Toc510704202)

[3.4 Felhasználó név és jelszó beállítása 11](#_Toc510704203)

[4. Szolgáltatás Smoke tesztelése 13](#_Toc510704204)

[4.1 UploadDocument operáció tesztelése 13](#_Toc510704205)

[4.1.1 Kísérő meta adatok 14](#_Toc510704206)

[4.1.2 Meta adat hiba 16](#_Toc510704207)

[4.2 GetDocumentStatus operáció 16](#_Toc510704208)

[4.3 GetDocumentErrorLog operáció tesztelése 17](#_Toc510704209)

[5. Későbbi önkormányzati fejlesztések 18](#_Toc510704210)

# Bevezetés

## Jelen dokumentum célja

Jelen dokumentum célja, hogy az adattárház smoke szolgáltatásának SoapUI-al történő használatát bemutassa. A SoapUI segítségével a smoke szolgáltatáson keresztül feltöltve egy állományt, tesztelhető az adattárház és az önkormányzat közötti kommunikációs csatorna (ez a Smoke teszt). A SoapUI csak a Smoke teszt során használandó, segítségével biztosítható, hogy az önkormányzatok fejlesztési feladatok végrehajtása nélkül tesztelhessék le a kommunikációs csatornát.

A teszt elvégzéségez megfelelő informatikai előképzettség szükséges.

A SoapUI eszközzel történő meghíváshoz szükséges:

* egy előre elkészített **DWHSmokeService-soapui-project.xml** projekt, amit be kell importálni az eszközbe. A projekt mintaként szolgál a kiküldendő kérések formátumára.
* egy **URL**, amellyel a szolgáltatás az EIP (Egységes Integrációs Platform) -en keresztül elérhető. Ez később kiküldésre kerül, ezt kell majd a 3.2 fejezetben bemutatott módon beállítani.

# Tesztelő eszköz

## SoapUI

A SoapUI egy ingyenes és nyílt forráskódú funkcionális tesztelési megoldás, mely könnyen használható grafikus interfésszel, és előre gyártott funkciókkal van ellátva. A SoapUI segítségével könnyen és gyorsan végezhető el webszolgáltatások tesztelése.

Az eszközben XML formátumban megjelennek a kiküldendő kérések, amelyek szerkeszthetők is, a kérés elküldése után a kapott válasz is megjelenik a kérés mellett.

Az alábbi oldalról letölthető (Open Source verzió-t kell letölteni):

<https://www.soapui.org/downloads/latest-release.html>

Az eszköz egyszerűen telepíthető a telepítés során csak a telepítés helyét lehet érdemes megváltoztatni, valamint a telepítendő komponensek közül elegendő csak a SoapUI komponenst telepíteni, ahogy az alábbi ábra mutatja.



1. ábra: Telepítendő komponensek

# Smoke tesztelés előtt elvégzendő feladatok

## Projekt importálás

Sikeres telepítés után a kapott projekt a fájl menü **Import Project** parancsának segítségével importálható be.



2. ábra: Projekt importálása

Importálás után az új projekt megjelenik a felület bal oldalán található navigátor panelen. A projektet kinyitva megjelennek az elérhető operációk (részletes leírás a 2.6, 2.7 és 2.8 fejezetekben) és a hozzájuk tartozó minta kérések (Request1). A minta kérések előre elkészített kérések, amelyek egy-egy példát tartalmaznak a szolgáltatás hívásakor megadandó paraméterekre.



3. ábra: Importált projekt és a benne lévő operációk és minta kérések

## Szolgáltatás címének megadása

A projekt importálása után be kell állítani a szolgáltatás címét, erre a projektben lévő **DWHSmokeServiceBinding-ra** duplán kattintva megjelenő felület **Service Endpoints** fülén van lehetőség.



4. ábra: DWHSmokeServiceBinding-ra duplán kattintva megjelenő felület

A megjelenő felületen lehetőség van új végpont felvételére is, de célszerűbb a már meglévőt módosítani, amelyhez elég csak belekattintani az **Endpoint** oszlopba, és ott átírni a kapott EIP-es elérésre (URL), ahogy az a következő ábrán látható.



5. ábra: Service Endpoints felület

## Minta kérések futtatása

A projektben található minta kérések futtatásához először a kérést meg kell nyitni. Ez történhet a kérés nevére duplán kattintva, illetve jobb egérgombbal a kérés nevére kattintva a **Show Request Editor** parancsot választva.

A kérést megnyitva megjelenik a kérés szerkesztő felület, amely két részre oszlik, a bal oldali felületen látható a kiküldendő kérés, a jobb oldalon pedig az elküldött kérésre kapott válasz.



6. ábra: Kérés szerkesztő

Kérés elküldése történhet a szerkesztőbe kattintva **ALT+ENTER** billentyű kombinációval, illetve a szerkesztő bal felső sarkában található zöld háromszögre kattintva. A kérés elküldése után a válasz a jobb oldali panelen fog megjelenni.



7. ábra: Kérés megnyitása

## Felhasználó név és jelszó beállítása

A szolgáltatás meghívásához kapott authentikációs adatokat (felhasználónév, jelszó) a SoapUI projektben található kérések első meghívása előtt kell beállítani az alábbi leírásnak megfelelően.

Első lépésként a kérés szerkesztő megnyitása után az ablak alján található, illetve az alábbi képen látható **Auth** gombra kell kattintani:



8. ábra: Auth fül

A gombra kattintás után egy új panel jelenik meg a szerkesztő alatt, ahogy ez az alábbi képen látható:



9. ábra: Auth panel

A panelen az Authorization legördülő listából az alábbi képen is látható **Add New Authorization** parancsot kell választani.



10. ábra: Authorization legördülő lista

A parancs kiválasztása után megjelenő ablakban a type legördülő listából a **Basic** opciót kell választani, ahogy az alábbi képen is látható.



11. ábra: Authorization típus

A Basic kiválasztása után az Auth panel megjelenő, az alábbi képen is látható **Username** és **Password** mezőben kell megadni a kapott felhasználói adatokat.



12. ábra: Felhasználónév és jelszó mezők

A felhasználó név és jelszó megadására szolgáló mezők alatt található egy **Pre-emptive auth** rádió gomb, ennek az értékét **Authenticate pre-emptively-**rekell állítani, ahogy az alábbi képen is látható.



. ábra: Pre-emptive auth rádió gomb

# Szolgáltatás Smoke tesztelése

## UploadDocument operáció tesztelése

Állomány feltöltésre a szolgáltatásnak ez az operációja biztosít lehetőséget. Az operáció paraméterei két részre oszthatók, a kisérő meta adatokra (4.1.1 fejezet), illetve az átküldendő állományra.

Az importált projekt tartalmaz egy **gazd\_urlap\_123456\_123456\_20170101\_20171231.tar.gz**

állományt, ennek feltöltése az elvárt a smoke teszt elvégzéséhez, ezen semmilyen módosítás elvégzése nem szükséges.

A küldés előtt a kísérő meta adatok beállítása szükséges, amelyeket a kérés **UploadDocumentMetaData** mezője tartalmaz a kérés fejlécében.



14. ábra: Meta adatok

A minta kérésben szereplő meta adatok formailag megfelelőek, azonban legalább a pirCode-ot módosítani kell, hogy a teszt során egyértelműen azonosítható legyen, hogy kitől érkezett a fájl, és az esetleglegesen jelentkező hibák okának feltárása könnyebb legyen.

* Amennyiben a feladó egy önkormányzat úgy kérjük a saját PIR kódját írja be
* Amennyiben a feladó egy szállító, aki adott esetben több önkormányzat állományait is küldi majd a betöltési prototípus során, a smoke teszt fázisában elegendő, ha a listából csak egy, tetszőlegesen választott PIR számot ad meg, és csak egyszer küldi vissza a minta állományt.

### Kísérő meta adatok

A küldendő állomány mellett a meta adatok a következők:

| **Adat** | **Leírás** | **Mintasor** |
| --- | --- | --- |
| fajlNev | Adott fájl neve | gazd\_urlap\_123456\_123456\_20170101\_20171231.tar.gz |
| pirCode | Csomaghoz tartozó PIR szám | 123456 |
| adatkorAzon | Adatkör azonosító (lásd 2. táblázat **– Adatkör kódok**) | URLAP |
| fajlVerzio | Fájl verziója | 1.0 |
| deperszonVerzio | Deperszonalizáció verziója | 1.0 |
| feladasAzonosito | Feladás azonosítója (universally unique identifier) | c8b78d18-534c-4502-b5c3-7ed37d426acf |
| feladasIdopontja | Feladás időpontja | 2018-03-14T15:00:00 |
| levalogatsIdopontja | Leválogatás időpontja | 2017-01-01T15:00:00 |
| idoszakTol | Adattartalom időszakának kezdete | 2017-01-01 |
| idoszakIg | Adattartalom időszakának vége | 2017-12-31 |
| feladoRendszer | Feladó szakrendszer azonosítója (lásd 1. táblázat **– Szakrendszer kódok**) | GAZD |
| checksum | checksum (MD5) | 84dac0243d8fc7e4926e020ed4b05e74 |

A kérés elküldése után kapott válaszban sikeres feladás esetén két adat található.

| **Név** | **Leírás** |
| --- | --- |

|  |  |
| --- | --- |
| documentId | Dokumentum azonosító, ezzel az azonosítóval kérdezhető le a feladott állomány feldolgozásának állapota. |
| varhatoAllapotValtozasSec | A rendszer szerint várhatóan ennyi másodperc múlva érdemes lekérdezni a feldolgozás állapotát, ennyi idő elteltével várható állapot változás. |



15. ábra: Példa sikeres állomány feltöltésre

### Meta adat hiba

Hibás vagy hiányzó meta adat esetén a válaszban hiba üzenet fog megjelenni, amely tájékoztatja a kérőt arról, hogy pontosan melyik adattal van probléma.

A hiba pontos oka a válasz üzenet **UploadDocumentFault** tagjében jelenik meg.



16. ábra: Példa meta adat hibára

## GetDocumentStatus operáció

A szolgáltatásnak ez az operációja biztosít lehetőséget egy adott állomány feldolgozás állapotának lekérdezésére.

A szolgáltatás meghívásához csak egy **documentId** paramétert szükséges átadni, amely meg kell egyezzen egy sikeres feltöltés során kapott válaszban szereplő azonosítóval.

Sikeres kérés elküldése esetén a válaszban található **status** mező tartalmazza a feldolgozás aktuális állapotát.

 

17. ábra: Státusz lekérdezés példa

##  GetDocumentErrorLog operáció tesztelése

A szolgáltatásnak ez az operációja biztosít lehetőséget a feldolgozás során keletkezett esetleges hiba log állományok lekérdezésére.

A szolgáltatás meghívásához csak egy **documentId** paramétert szükséges átadni, amely meg kell egyezzen egy sikeres feltöltés során kapott válaszban szereplő azonosítóval.

A válaszban a **fileName** tartalmazza a hiba állomány nevét, a **fileContent** pedig magát a hiba állományt.



18. ábra: Hiba log állomány lekérdezés példa

Amennyiben egy állományhoz nem tartozik hiba állomány, abban az esetben a válasz egy „Hiba log keresési hiba” üzenetet fog tartalmazni.

# Későbbi önkormányzati fejlesztések

A SoapUI csak a smoke teszt során használandó, segítségével biztosítható, hogy az önkormányzatok fejlesztési feladatok végrehajtása nélkül tesztelhessék le a kommunikációs csatornát.

A sikeres teszteléseket követően az önkormányzatoknak/szállítóknak a későbbiekben, a kapott projektben lévő WSDL alapján programozottan kell az állományokat feladniuk. A programozott megvalósításban az állományának mellett küldendő kisérő meta adatok dinamikus megadhatósága ajánlott, az egyes adatok statikus megadásával szemben.

A kialakítandó interfészről, az ütemezetten küldendő fájlok formátumáról és tartalmáról a szintén elküldött **ASP2 0\_DWH\_Interfész\_specifikáció\_v1.0.docx** dokumentum tartalmaz részletes leírásokat. (a mintafájl elnevezése is ennek megfelelően lett kialakítva, csak valós PIR-szám helyett az 123456 értéket tartalmazza)