

SoapUI Házi Feladat

Teszt Automatizálás Alapok

Tartalom

Feladat: Teszt projekt beállítása	3
Feladat: saját web szolgáltatás projekt létrehozása és első saját hívás futtatása	7
Feladat: saját web szolgáltatás projekt létrehozása és első saját hívás futtatása	7
Feladat: A bemutatott módon készíts tesztek és adj hozzájuk asserteket.....	8
Feladat: saját web szolgáltatás projekt létrehozása és első saját hívás futtatása	9
Feladat: saját web szolgáltatás projekt létrehozása és első saját hívás futtatása	10
Feladat: Feladat beadása github segítségével.	11

Feladat: Teszt projekt beállítása

Előfeltételek:

- Előzőleg már telepítetted a Docker környezetet a gépedre
- Előzőleg már telepítetted a git alkalmazást a gépedre

A feladatot az lesz, hogy a locations-app teszt alkalmazásra felületi automatizált teszteket készíts. Ezt az alkalmazást az alábbi github repoban találod: <https://github.com/Training360/locations-app>

Az alkalmazást forrását az alábbi paranccsal tudod le klónizni:

Parancssorban:

git clone <https://github.com/Training360/locations-app.git>

(A másik lehetőség, hogy letöltöd a Github oldalról az egész repozitorit egy ZIP fileként.)

A locations-app mappában található egy locations almappa. Ebben van egy docker-compose.yml file.

Ez tartalmazza a különböző paramétereket a konténerek elindításához.

Az alábbi paranccsal el tudod indítani a környezetet a saját gépeden:

Parancssorban:

cd locations-app

cd locations

docker-compose up -d

Ha bármilyen gond akad (lássd megosztott gyakori problémák leírást) akkor érdemes

docker-compose up

(a -d kapcsoló nélkül)

paranccsal indítani és megnézni a logokat. Ekkor ctrl + c-vel lehet kilépni és leállítani a futást.

```
Command Prompt
C:\Users\tjozs\projects\git\github\tjozs\locations-app\locations>docker-compose up -d
Starting locations_locations-mariadb_1 ... done
Starting locations_locations-app_1      ... done

C:\Users\tjozs\projects\git\github\tjozs\locations-app\locations>_
```

Az alkalmazás kisvártatva elindul és egy új böngésző ablakban a localhost 8080-as portján érhető el.

Web böngészőben nyissd meg az alábbi url-t.

<http://localhost:8080>

Locations

localhost:8080

Locations

[Create location](#)

Refresh table

Id	Name	Coordinates	Interesting at	Tags	
1	Budapest	47.497912, 19.040235	2019-09- 17T05:00:00	capital,favourite	Edit Delete
20	Budapest	47.497912, 19.040235	2021-02- 02T04:44:54	capital,favourite	Edit Delete
24	Budapest	47.497912, 19.040235	2021-02- 02T04:44:54	capital,favourite	Edit Delete
8	Budapest	47.497912, 19.040235	2021-02- 02T04:44:54	capital,favourite	Edit

Ez a projekt elérhetővé tesz több különböző SAOP web szolgáltatást. Ezeket a weblap alján lévő linken tudod megnézni:

29	Washington D.C.	38.907192, -77.036873	2019-09- 11T15:31:04	capital	Edit Delete
----	--------------------	--------------------------	-------------------------	---------	--

[SOAP webservice](#)

[Swagger](#)

[OpenAPI](#)

REST webservice [API documentation](#)

Másik lehetőség, hogy direktben megnyitod az alábbi url-t: <http://localhost:8080/services>

CXF - Service list	
localhost:8080/services	
Available SOAP services:	
AuthEndpoint <ul style="list-style-type: none"> sayHello 	Endpoint address: http://localhost:8080/services/auth WSDL : http://localhost:8080/services/locations?wsdl Target namespace: http://locations.com/services/locations
FilesEndpoint <ul style="list-style-type: none"> uploadFile downloadFile downloadFileAsBase64 	Endpoint address: http://localhost:8080/services/files WSDL : http://localhost:8080/services/locations?wsdl Target namespace: http://locations.com/services/locations
LocationsEndpoint <ul style="list-style-type: none"> updateLocation deleteLocation createLocation listLocations 	Endpoint address: http://localhost:8080/services/locations WSDL : http://localhost:8080/services/locations?wsdl Target namespace: http://locations.com/services/locations

A feladatot a LocationsEndpoint SOAP webservice tesztelése lesz. Ehhez a SWDL linkre kattintva vagy az alábbi WSDL urlt másolva tudod a SoapUI projektet bekonfigurálni.

<http://localhost:8080/services/locations?wsdl>

localhost:8080/services/locations

localhost:8080/services/locations?wsdl

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tns="http://locations.com/services/locations" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:ns1="http://schemas.xmlsoap.org/soap/http" name="LocationsEndpointService"
  targetNamespace="http://locations.com/services/locations">
  <wsdl:types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://locations.com/services/locations"
      elementFormDefault="qualified" targetNamespace="http://locations.com/services/locations" version="1.0">
      <xs:element name="createLocation" type="tns:createLocation"/>
      <xs:element name="createLocationResponse" type="tns:createLocationResponse"/>
      <xs:element name="deleteLocation" type="tns:deleteLocation"/>
      <xs:element name="deleteLocationResponse" type="tns:deleteLocationResponse"/>
      <xs:element name="listLocations" type="tns:listLocations"/>
      <xs:element name="listLocationsResponse" type="tns:listLocationsResponse"/>
      <xs:element name="updateLocation" type="tns:updateLocation"/>
      <xs:element name="updateLocationResponse" type="tns:updateLocationResponse"/>
      <xs:complexType name="updateLocation">
        <xs:sequence>
          <xs:element name="updateLocationRequest" type="tns:updateLocationRequest"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </wsdl:types>
  <wsdl:port name="LocationsEndpointService" type="tns:LocationsEndpointService" binding="tns:LocationsEndpointService">
    <soap:address location="http://localhost:8080/services/locations">
    </soap:address>
  </wsdl:port>
</wsdl:definitions>

```

Feladat: saját web szolgáltatás projekt létrehozása és első saját hívás futtatása

Előfeltételek:

- Előzőleg már elindítottad a Teszt Projektet a saját gépeden Docker segítségével
- Telepítettet a SoapUI appot

Nissd meg megnyitod az alábbi url-t: <http://localhost:8080/services>

A feladatot a LocationsEndpoint SOAP webservice tesztelése lesz. Ehhez a SWDL linkre kattintva vagy az alábbi WSDL urlt másolva tudod a SoapUI projektet bekonfigurálni.

<http://localhost:8080/services/locations?wsdl>

Feladatod: Készíts egy új SoapUI projektet a fenti WSDL alapján. A bemutatott módon az összes operation-re készíts egy-egy SOAP request-et az eszközzel. Hívd meg ezeket a requesteket és figyeld meg a működésüket.

Feladat: saját web szolgáltatás projekt létrehozása és első saját hívás futtatása

Előfeltételek:

- Előzőleg már elindítottad a Teszt Projektet a saját gépeden Docker segítségével
- Telepítettet a SoapUI appot
- Készítettél egy saját SoapUI projektet és a WSDL segítségével már létrehoztad az operation-ökre a futtatásokat (requests)

Például:

- Kérdezd le az összes jelenleg ismert lokációt
- Kérdezz le egy bizonyos lokációt
- Hozz létre egy lokációt és ellenőrizd, hogy tényleg létre jött-e
- Feladat: saját web szolgáltatás projekt létrehozása és első saját hívás futtatása

Feladat: A bemutatott módon készíts tesztek és adj hozzájuk asserteket.

Előfeltételek:

- Előzőleg már elindítottad a Teszt Projektet a saját gépeden Docker segítségével
- Telepítettél a SoapUI appot
- Készítettél egy saját SoapUI projektet és a WSDL segítségével már létrehoztad az operation-ökre a futtatásokat (requests)
- Készítettél már tesztek, amiket most ellenőrzéssel tudsz majd bővíteni

Feladatod: Adj hozzá a tesztjeidhez olyan asser-eket, ami azt validálják, hogy helyes SOAP válasz érkezett-e.

Feladat: saját web szolgáltatás projekt létrehozása és első saját hívás futtatása

Előfeltételek:

- Előzőleg már elindítottad a Teszt Projektet a saját gépeden Docker segítségével
- Telepítettél a SoapUI appot
- Készítettél egy saját SoapUI projektet és a WSDL segítségével már létrehoztad az operation-ökre a futtatásokat (requests)
- Készítettél már tesztek, amiket most ellenőrzéssel tudsz majd bővíteni

Feladatod: Bővítsd a validációdat XPath típusú validációval a bemutatott módon. Itt az általad megvalósított tesztnek megfelelő visszatérési értékre készüljön az XPath assert. Azt kell megvizsgálnia, hogy helyes eredményt adott-e vissza az adott kérés.

Feladat: saját web szolgáltatás projekt létrehozása és első saját hívás futtatása

Előfeltételek:

- Előzőleg már elindítottad a Teszt Projektet a saját gépeden Docker segítségével
- Telepítetted a SoapUI appot
- Készítettél egy saját SoapUI projektet és a WSDL segítségével már létrehoztad az operation-ökre a futtatásokat (requests)
- Készítettél már teszteket
- Vannak ellenőrzéseid

Feladatod: A beépített Test runner megmutatja, hogy milyen parancsot hajt végre. Ugyanezt a parancsot másold ki és futtasd le a parancssorban. A kimenetet mentsd le mert kelleni fog még.

Feladat: Feladat beadása github segítségével.

Az elkészített SoapUI Teszt Projektet mentsd el a lemezre. Az így keletkező xml file-t futtasd a command line runner-el. Ennek az eredményét is mentsd el egy file-ba. Mindekét file-t töltsd fel a feladathoz számodra létrehozott Github repoba.