

SOAP – Simple Object Access Protocol

SOAP je štandard, založený na XML správach. Je to protokol na výmenu správ medzi počítačmi.

- Je komunikačný protokol na komunikáciu cez internet
- Používa sa v http
- Používa sa pri prenose dát webových služieb
- Môže vymieňať celé dokumenty alebo len vzdialené procedúry
- Komunikácia prebieha väčšinou ako klient a server

Správa SOAP obsahuje prvky ako :

Obálka (začiatok a koniec správy), **hlavička** (voliteľný prvok, atribúty správy, ktoré potrebujeme pri spracovaní), **telo** (dáta XML), **chyby** (informácie o chybách).

Príklad komunikácie:

Ak chce klient zistiť podrobnejšie informácie o produkte:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetails xmlns="http://warehouse.example.com/ws">
      <productID>827635</productID>
    </getProductDetails>
  </soap:Body>
</soap:Envelope>
```

Čo odpovie webová služba:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetailsResponse xmlns="http://warehouse.example.com/ws">
      <getProductDetailsResult>
        <productName>Čokoláda sada 3 chutí</productName>
        <productID>827635</productID>
        <popis>Čokoláda horká, biela a smotanová</popis>
        <cena>98,50</cena>
        <naSklade>áno</naSklade>
      </getProductDetailsResult>
    </getProductDetailsResponse>
  </soap:Body>
</soap:Envelope>
```

INŠTALÁCIA:

Najprv si zo stránky: <https://tomcat.apache.org/download-80.cgi> stiahneme installer.

Ak nie je v počítači nainštalovaný balíček Eclipse IDE for Java EE Developers treba si ho stiahnuť zo stránky : <https://eclipse.org/downloads/eclipse-packages/>

TomCat aj Eclipse (ak v pc nie je) nainštalujeme.

Po otvorení Eclipse prejdeme do **window -> Preferences -> Server -> Runtime Environments -> Add**

Tu pridáme náš **Apache Tomcat(v8.0)**, ktorý sme stiahli.

Musíme tam nastaviť cestu kde sme to inštalovali: **C: -> Program Files -> Apache Software Foundation -> TomCat 8.0**

Vytváranie webovej služby SOAP a klientský program pomocou Eclipse:

Budeme pracovať s Apache Axis, ktorý je integrovaný v Eclipse. Axis poskytuje rýchly a jednoduchý spôsob ako meniť aplikácie do Java Web Service na testovanie.

1. Najprv si v Eclipse vytvoríme nový **Dynamic Web Project**.
2. Pomenujeme si náš projekt napr. SOAPExample.

Vytvorili sme si projekt, ktorý teraz ideme modifikovať. Povedzme, že chceme aby naša webová služba vedela pridávať, mazať a získavať objekty.

V **src** si vytvoríme **package : com.journaldev.jaxws.beans**

V ňom triedu **Person**, do ktorého vložíme tento text:

```
package com.journaldev.jaxws.beans;

import java.io.Serializable;

public class Person implements Serializable{

    private static final long serialVersionUID = -
5577579081118070434L;

    private String name;

    private int age;

    private int id;
```

```

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public int getAge() {
    return age;
}

public void setAge(int age) {
    this.age = age;
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

@Override
public String toString(){
    return id+"::"+name+"::"+age;
}
}

```

Ďalším krokom bude vytvoriť **service classes**, ktoré bude mať rozhranie **Person Services**.

Vytvoríme si **package** : **com.journaldev.jaxws.service**

V ňom vytvoríme **interface PersonService** a vložíme tam nasledujúci kód:

```

package com.journaldev.jaxws.service;

import com.journaldev.jaxws.beans.Person;

public interface PersonService {

    public boolean addPerson(Person p);

    public boolean deletePerson(int id);

    public Person getPerson(int id);

    public Person[] getAllPersons();

}

```

Do toho istého balíčka vytvoríme triedu **PersonServiceImpl** a vložíme kód:

Ak by sme niečím takým pracovali naozaj, bolo by lepšie na tieto informácie použiť databázu ako **Map**.

```

package com.journaldev.jaxws.service;

import java.util.HashMap;
import java.util.Map;
import java.util.Set;
import com.journaldev.jaxws.beans.Person;

public class PersonServiceImpl implements PersonService {

    private static Map<Integer, Person> persons = new
    HashMap<Integer, Person>();

    @Override

    public boolean addPerson(Person p) {

        if(persons.get(p.getId()) != null) return false;

        persons.put(p.getId(), p);

        return true;

    }

    @Override

    public boolean deletePerson(int id) {

```

```

        if(persons.get(id) == null) return false;

        persons.remove(id);

        return true;

    }

    @Override

    public Person getPerson(int id) {

        return persons.get(id);

    }

    @Override

    public Person[] getAllPersons() {

        Set<Integer> ids = persons.keySet();

        Person[] p = new Person[ids.size()];

        int i=0;

        for(Integer id : ids){

            p[i] = persons.get(id);

            i++;

        }

        return p;

    }

}

```

Keďže už máme hotovú „business logic“ je na čase vytvoriť Webovú službu. Teda klikneme na **New** a nájdeme **Web Service**, stlačíme next.

Teraz prichádza najdôležitejšia časť, teda kde webovej službe a klientovi poskytujeme potrebné údaje. Musíme sa uistiť, či v sekcii Web Service Type je **Bottom up Java bean Web Service**. Musíme vybrať toto, pretože my sme si najprv vytvorili implementáciu a teraz z nej generujeme súbor **WSDL**. V prvej konfigurácii musíme nastaviť **Test service**, teda potiahneme scroller až na vrch. To isté spravíme aj pri konfigurácii klienta.

Teda budeme mať Test service a Test Client.

Tlačidlo Next. Opäť Next.

Dostali sme sa k bodu kedy chceme zapnúť Server. Teda klikneme na **Start Server**. Dáme **Test Web service Explorer**. **Web Service Proxy** necháme tak ako je a dáme NEXT. V nasledujúcom okne, môžeme vidieť, aké metódy naša mala aplikácia ponúka. Tu nemeňte nič, dáme **FINISH**. Teraz nám naskočí okno prehliadača, kde si môžeme vyskúšať naše webové služby, ktoré sme vytvorili.

Môžeme si tiež skopírovať URL adresu a vložiť ju do prehliadača, aby sme sa presvedčili, že náš server vážne beží a funguje.

Úloha 1:

Skúsme si pridať niekoľko osôb. Potom vypísať jednu osobu. Nakoniec niektorú(všetky) osobu(y) zmazať.

Úloha 2:

Skúsme vytvoriť jednoduchú aplikáciu, podľa toho ako sme to robili teraz. Vytvoríte projekt rovnako ako tento s Osobami. Názov projektu je na vás. V **src** nemusíte vytvárať viac **Package**. Stačí jeden, ktorý nazvete **org.arpit.javapostsforlearning.webservices**. V ňom vytvorte triedu **HelloWorld**.

V triede nech je:

```
package org.arpit.javapostsforlearning.webservices;

public class HelloWorld {

    public String sayHelloWorld(String name)
    {
        return "Hello world from "+ name;
    }
}
```

Vytvorte **web service**. Postup nájdete hore. Nakoniec to spustíte v prehliadači.