

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

канд. техн. наук, доцент

должность, уч. степень, звание

подпись, дата

А. В. Фомин

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

РЕАЛИЗАЦИЯ SOAP ВЕБ-СЕРВИСА

по курсу: СОВРЕМЕННЫЕ ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4232М

подпись, дата

В. Ф. Губайдулин

инициалы, фамилия

Санкт-Петербург 2023

Цель работы:

Изучение средств Java для разработки web сервисов.

Вариант:

Разрабатываемое приложение – автоматизированная система управления составом футбольной команды.

Система будет позволять работать со списком команд, с игроками, играющими за определённую команду, а также с формациями – схемами игры.

Ход работы:

1) SOAP (Simple Object Access Protocol) — протокол обмена структурированными сообщениями в распределённой вычислительной среде.

Сообщение SOAP выглядит так:

- Envelope — корневой элемент, который определяет сообщение и пространство имен, использованное в документе.
- Header — содержит атрибуты сообщения, например: информация о безопасности или о сетевой маршрутизации.
- Body — содержит сообщение, которым обмениваются приложения.
- Fault — необязательный элемент, который предоставляет информацию об ошибках, которые произошли при обработке сообщений.

WSDL (Web Services Description Language) — язык описания веб-сервисов и доступа к ним, основанный на языке XML.

2) Был реализован класс WSConfig, который определяет Endpoint'ы для сервисов.

Листинг 1 – Класс WSConfig

```
@Configuration
public class WSConfig {

    @Autowired
    private Bus bus;
    @Autowired
    private IPlayerRepos _playerRepos;
    @Autowired
    private ISkillsRepos _skillsRepos;
```

```

@Bean
public Endpoint helloEndpoint() {
    EndpointImpl endpoint = new EndpointImpl(bus, new HelloServiceImpl());
    endpoint.publish("/Hello");
    return endpoint;
}

@Bean
public Endpoint addSkillsEndpoint() {
    EndpointImpl endpoint = new EndpointImpl(bus, new
SkillsServiceImpl(_playerRepos, _skillsRepos));
    endpoint.publish("/Skills");
    return endpoint;
}
}

```

3) Были реализованы сервисы и их интерфейсы: `HelloService`, `SkillsService`. Реализация класса `SkillsServiceImpl` представлен в приложении А. Полученный WSDL файл представлен в приложении Б. В классе `SkillsServiceImpl` реализованы следующие методы:

- `addOrUpdSkills(SkillsDto skills)` – реализует добавление или обновление критериев навыков для игроков в зависимости от года начала сезона.
- `getSkills(Integer playerId, Integer season)` – реализует получение критериев навыков игрока в зависимости от года начала сезона.
- `delSkills(Integer playerId, Integer season)` – реализует удаление критериев навыков для игроков в зависимости от года начала сезона.

Листинг 2 – Реализация `SkillsService`

```

@WebService(name = "addSkill")
public interface SkillsService {

    @WebMethod(operationName = "addSkill")
    String addOrUpdSkills(@WebParam(name="skills", targetNamespace = "")
SkillsDto skills) throws SoapException;

    @WebMethod(operationName = "getSkill")
    Skills getSkills(@WebParam(name="playerId", targetNamespace = "") Integer
playerId,
                    @WebParam(name="season", targetNamespace = "") Integer
season) throws SoapException;

    @WebMethod(operationName = "delSkill")
    String delSkills(@WebParam(name="playerId", targetNamespace = "") Integer
playerId,
                    @WebParam(name="season", targetNamespace = "") Integer
season) throws SoapException;
}

```

4) Вышеописанные методы были протестированы с помощью SoapUI.

Available SOAP services:

addSkill <ul style="list-style-type: none">addSkillgetSkilldelSkill	Endpoint address: http://localhost:8080/Service/Skills WSDL : http://soap.footballers.example.com/Skills Target namespace: http://soap.footballers.example.com/
sayHello <ul style="list-style-type: none">hello	Endpoint address: http://localhost:8080/Service/Hello WSDL : http://soap.footballers.example.com/Hello Target namespace: http://soap.footballers.example.com/

Рисунок 1 – Список доступных SOAP сервисов

The screenshot shows the SoapUI interface. The top bar indicates 'Request 1' and the endpoint 'http://localhost:8080/Service/Skills'. The left pane shows the raw XML of the request, and the right pane shows the raw XML of the response. Below the XML view is a 'Result Grid' table.

	id	defence	dribbling	passing	physic	player	season_start	shot	speed
▶	43	7	8	5	4	1	2022	8	9
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 2 – Тест метода addOrUpdSkills

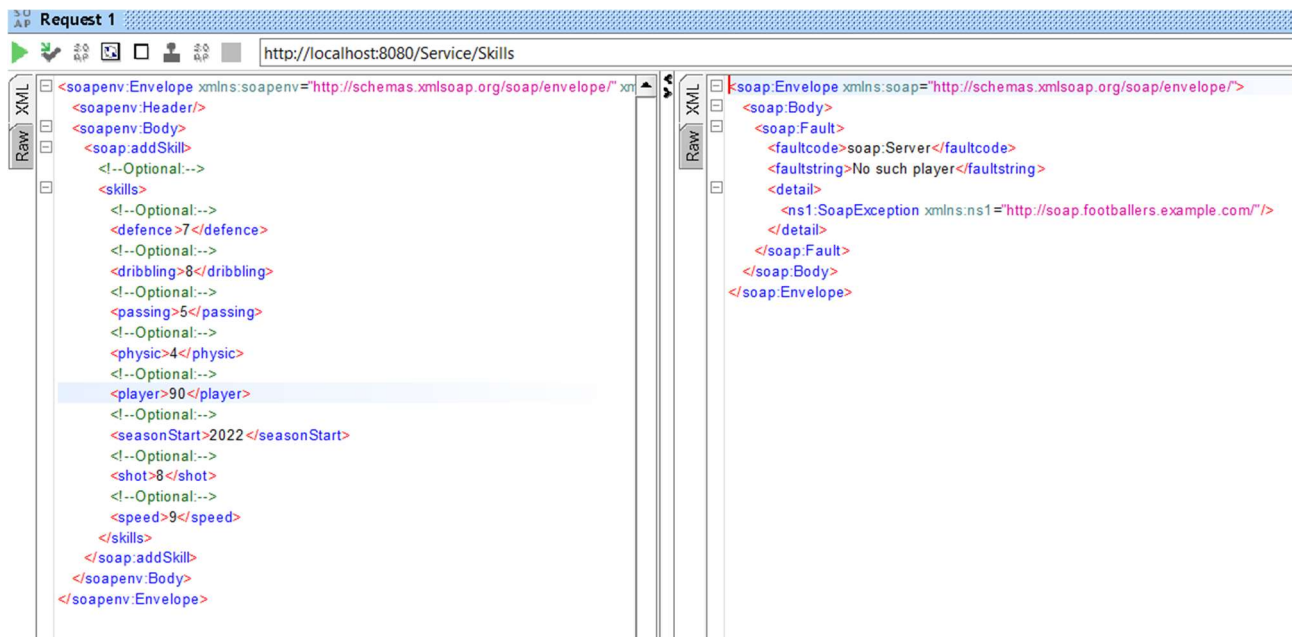


Рисунок 3 – Тест метода addOrUpdSkills если нет игрока с введённым id

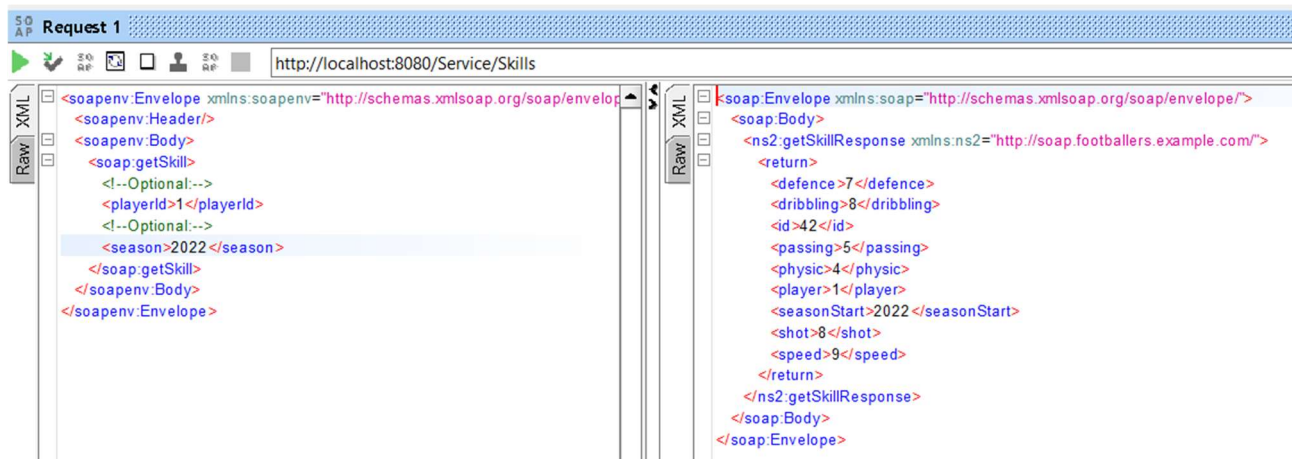


Рисунок 4 – Тест метода getSkills

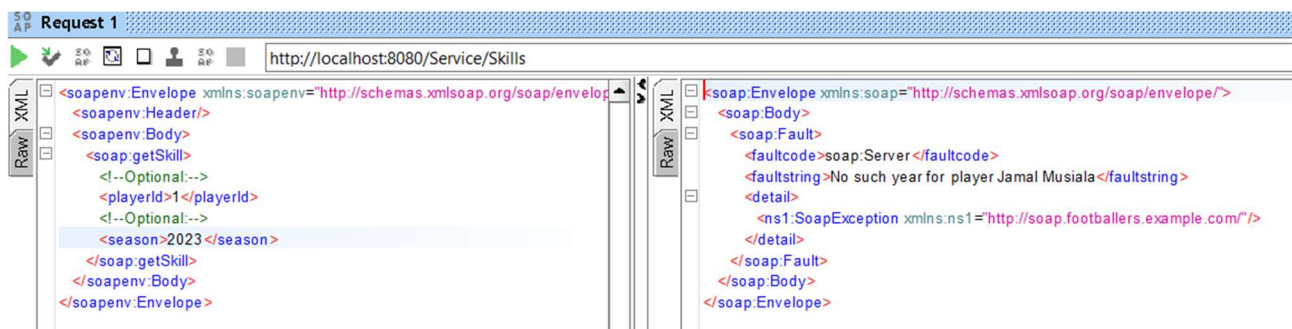


Рисунок 5 – Тест метода getSkills при неверно введённом годе

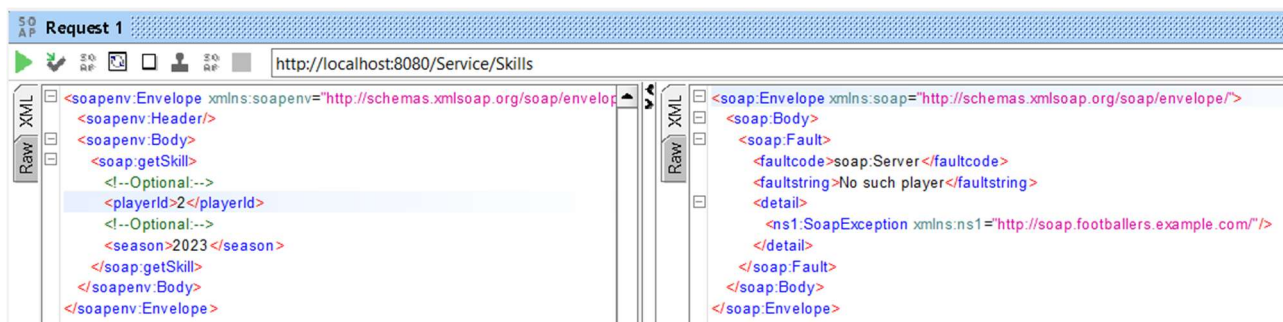


Рисунок 6 – Тест метода getSkills при неверно введённом id игрока

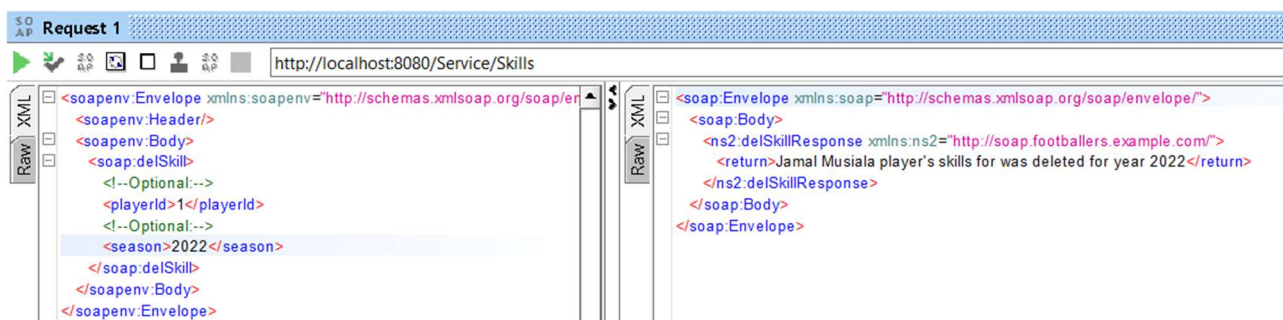


Рисунок 7 – Тест метода delSkills

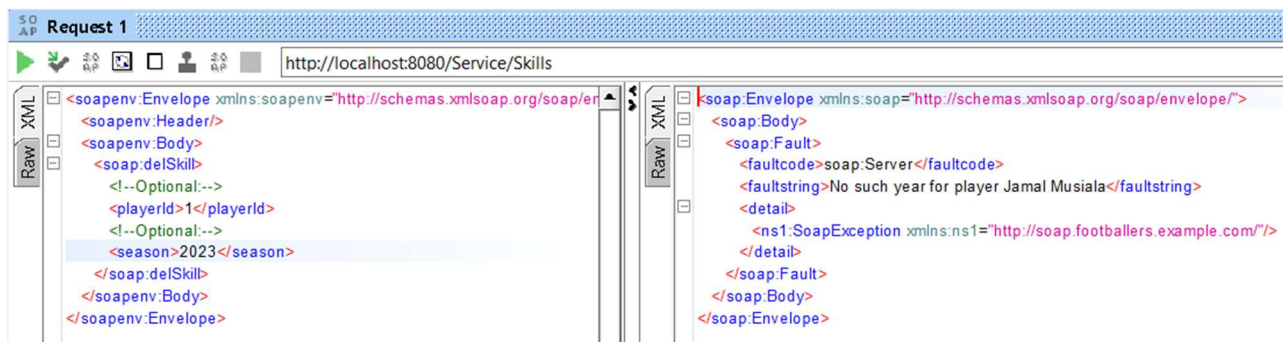


Рисунок 8 – Тест метода delSkills при неверно введённом годе

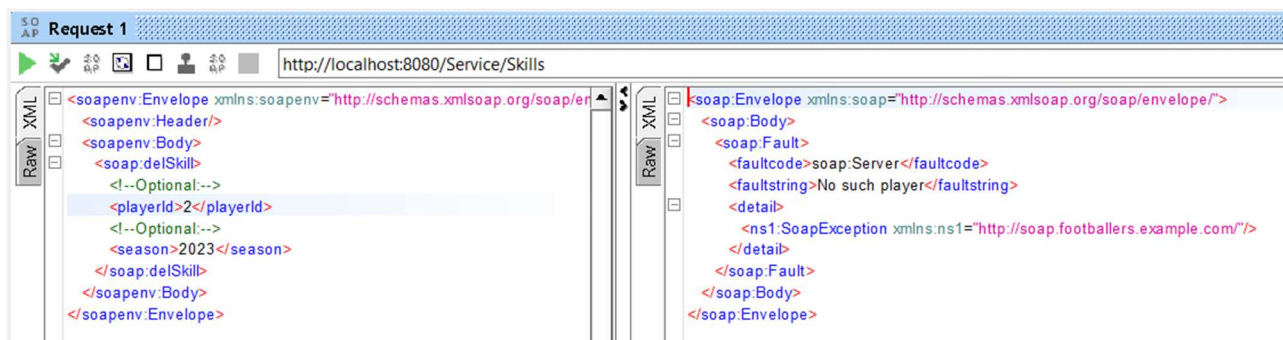


Рисунок 9 – Тест метода delSkills при неверно введённом id игрока

Вывод:

Были изучены средства Java для разработки web сервисов. Были реализованы и протестированы методы, использующиеся web сервисом.

Приложение А. Листинг класса SkillsServiceImpl

```
@WebService(serviceName = "Skills")
public class SkillsServiceImpl implements SkillsService {

    private final IPlayerRepos _playerRepos;
    private final ISkillsRepos _skillsRepos;

    public SkillsServiceImpl(IPlayerRepos playerRepos, ISkillsRepos
skillsRepos){
        _playerRepos = playerRepos;
        _skillsRepos = skillsRepos;
    }

    @Override
    public String addOrUpdSkills(SkillsDto skills) throws SoapException {
        String result = "{0} player's skills was updated";
        Players player = _playerRepos.findById(skills.getPlayer()).orElse(null);
        if (player == null) {
            Throwable t = new NoSuchElementException("No such player");
            throw new SoapException("No such player", t);
        }
        Skills existedSkills =
_skillsRepos.getSkillsByPlayerAndSeason(skills.getPlayer(),
skills.getSeasonStart());
        if (existedSkills == null) {
            Skills skillsToAdd = SkillsMapper.INSTANCE.toModel(skills);
            _skillsRepos.save(skillsToAdd);
        } else {
            if (skills.getDefence() != null)
                existedSkills.setDefence(skills.getDefence());
            if (skills.getPassing() != null)
                existedSkills.setPassing(skills.getPassing());
            if (skills.getDribbling() != null)
                existedSkills.setDribbling(skills.getDribbling());
            if (skills.getPhysic() != null)
                existedSkills.setPhysic(skills.getPhysic());
            if (skills.getShot() != null)
                existedSkills.setShot(skills.getShot());
            if (skills.getSpeed() != null)
                existedSkills.setSpeed(skills.getSpeed());
            _skillsRepos.save(existedSkills);
        }
        return MessageFormat.format(result, player.getName());
    }

    @Override
    public Skills getSkills(Integer playerId, Integer season) throws
SoapException {
        Players player = _playerRepos.findById(playerId).orElse(null);
        if (player == null) {
            Throwable t = new NoSuchElementException("No such player");
            throw new SoapException("No such player", t);
        }
        Skills existedSkills = _skillsRepos.getSkillsByPlayerAndSeason(playerId,
season);
        if (existedSkills == null) {
            Throwable t = new NoSuchElementException("No such year for player "
+ player.getName());
            throw new SoapException("No such year for player " +
player.getName(), t);
        }
    }
}
```

```

        return existedSkills;
    }

    @Override
    public String delSkills(Integer playerId, Integer season) throws
SoapException {
        String result = "%s player's skills for was deleted for year %s";
        Players player = _playerRepos.findById(playerId).orElse(null);
        if (player == null) {
            Throwable t = new NoSuchElementException("No such player");
            throw new SoapException("No such player", t);
        }
        Skills existedSkills = _skillsRepos.getSkillsByPlayerAndSeason(playerId,
season);
        if (existedSkills == null) {
            Throwable t = new NoSuchElementException("No such year for player "
+ player.getName());
            throw new SoapException("No such year for player " +
player.getName(), t);
        }
        _skillsRepos.delete(existedSkills);
        return String.format(result, player.getName(), season.toString());
    }
}

```


Приложение Б. WSDL файл для SkillsService

```
<wsdl:definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://soap.footballers.example.com/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:ns1="http://schemas.xmlsoap.org/soap/http" name="Skills" targetNamespace="http://soap.footballers.example.com/">
  <wsdl:types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://soap.footballers.example.com/" attributeFormDefault="unqualified" elementFormDefault="unqualified" targetNamespace="http://soap.footballers.example.com/" version="1.0">
      <xs:element name="addSkill" type="tns:addSkill"/>
      <xs:element name="addSkillResponse" type="tns:addSkillResponse"/>
      <xs:element name="delSkill" type="tns:delSkill"/>
      <xs:element name="delSkillResponse" type="tns:delSkillResponse"/>
      <xs:element name="getSkill" type="tns:getSkill"/>
      <xs:element name="getSkillResponse" type="tns:getSkillResponse"/>
      <xs:complexType name="addSkill">
        <xs:sequence>
          <xs:element minOccurs="0" name="skills" type="tns:skillsDto"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="skillsDto">
        <xs:sequence>
          <xs:element minOccurs="0" name="defence" type="xs:int"/>
          <xs:element minOccurs="0" name="dribbling" type="xs:int"/>
          <xs:element minOccurs="0" name="passing" type="xs:int"/>
          <xs:element minOccurs="0" name="physic" type="xs:int"/>
          <xs:element minOccurs="0" name="player" type="xs:int"/>
          <xs:element minOccurs="0" name="seasonStart" type="xs:int"/>
          <xs:element minOccurs="0" name="shot" type="xs:int"/>
          <xs:element minOccurs="0" name="speed" type="xs:int"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="addSkillResponse">
        <xs:sequence>
          <xs:element minOccurs="0" name="return" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="getSkill">
        <xs:sequence>
          <xs:element minOccurs="0" name="playerId" type="xs:int"/>
          <xs:element minOccurs="0" name="season" type="xs:int"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="getSkillResponse">
        <xs:sequence>
          <xs:element minOccurs="0" name="return" type="tns:skills"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="skills">
        <xs:sequence>
          <xs:element minOccurs="0" name="defence" type="xs:int"/>
          <xs:element minOccurs="0" name="dribbling" type="xs:int"/>
          <xs:element minOccurs="0" name="id" type="xs:int"/>
          <xs:element minOccurs="0" name="passing" type="xs:int"/>
          <xs:element minOccurs="0" name="physic" type="xs:int"/>
          <xs:element minOccurs="0" name="player" type="xs:int"/>
          <xs:element minOccurs="0" name="seasonStart" type="xs:int"/>
          <xs:element minOccurs="0" name="shot" type="xs:int"/>
          <xs:element minOccurs="0" name="speed" type="xs:int"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </wsdl:types>

```

```

</xs:complexType>
<xs:complexType name="delSkill">
<xs:sequence>
<xs:element minOccurs="0" name="playerId" type="xs:int"/>
<xs:element minOccurs="0" name="season" type="xs:int"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="delSkillResponse">
<xs:sequence>
<xs:element minOccurs="0" name="return" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:element name="SoapException" type="tns:SoapException"/>
<xs:complexType name="SoapException">
<xs:sequence>
<xs:element minOccurs="0" name="message" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
</wsdl:types>
<wsdl:message name="addSkill">
<wsdl:part element="tns:addSkill" name="parameters"> </wsdl:part>
</wsdl:message>
<wsdl:message name="getSkillResponse">
<wsdl:part element="tns:getSkillResponse" name="parameters"> </wsdl:part>
</wsdl:message>
<wsdl:message name="SoapException">
<wsdl:part element="tns:SoapException" name="SoapException"> </wsdl:part>
</wsdl:message>
<wsdl:message name="getSkill">
<wsdl:part element="tns:getSkill" name="parameters"> </wsdl:part>
</wsdl:message>
<wsdl:message name="delSkill">
<wsdl:part element="tns:delSkill" name="parameters"> </wsdl:part>
</wsdl:message>
<wsdl:message name="addSkillResponse">
<wsdl:part element="tns:addSkillResponse" name="parameters"> </wsdl:part>
</wsdl:message>
<wsdl:message name="delSkillResponse">
<wsdl:part element="tns:delSkillResponse" name="parameters"> </wsdl:part>
</wsdl:message>
<wsdl:portType name="addSkill">
<wsdl:operation name="addSkill">
<wsdl:input message="tns:addSkill" name="addSkill"> </wsdl:input>
<wsdl:output message="tns:addSkillResponse" name="addSkillResponse"> </wsdl:output>
<wsdl:fault message="tns:SoapException" name="SoapException"> </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="getSkill">
<wsdl:input message="tns:getSkill" name="getSkill"> </wsdl:input>
<wsdl:output message="tns:getSkillResponse" name="getSkillResponse"> </wsdl:output>
<wsdl:fault message="tns:SoapException" name="SoapException"> </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="delSkill">
<wsdl:input message="tns:delSkill" name="delSkill"> </wsdl:input>
<wsdl:output message="tns:delSkillResponse" name="delSkillResponse"> </wsdl:output>
<wsdl:fault message="tns:SoapException" name="SoapException"> </wsdl:fault>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="SkillsSoapBinding" type="tns:addSkill">

```

```

<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="addSkill">
<soap:operation soapAction="" style="document"/>
<wsdl:input name="addSkill">
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output name="addSkillResponse">
<soap:body use="literal"/>
</wsdl:output>
<wsdl:fault name="SoapException">
<soap:fault name="SoapException" use="literal"/>
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="getSkill">
<soap:operation soapAction="" style="document"/>
<wsdl:input name="getSkill">
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output name="getSkillResponse">
<soap:body use="literal"/>
</wsdl:output>
<wsdl:fault name="SoapException">
<soap:fault name="SoapException" use="literal"/>
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="delSkill">
<soap:operation soapAction="" style="document"/>
<wsdl:input name="delSkill">
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output name="delSkillResponse">
<soap:body use="literal"/>
</wsdl:output>
<wsdl:fault name="SoapException">
<soap:fault name="SoapException" use="literal"/>
</wsdl:fault>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="Skills">
<wsdl:port binding="tns:SkillsSoapBinding" name="SkillsServiceImplPort">
<soap:address location="http://localhost:8080/Service/Skills"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>

```