*Университет за национално и световно стопанство*

*Факултет: Приложна информатика и статистика*

*Специалност: Бизнес Информатика*



**Курсов проект**

По Интернет Технологии

Тема: **Web Service Definition Language (WSDL)**

Изготвил: **Теодор Иванов Куртев**

**Фак. № 118020**

**Група 1708, Поток 177**

София, 2014г.

Съдържание

[1. Същност 2](#_Toc387153508)

[2. Бръз преглед на концепцията за уеб услуги 2](#_Toc387153509)

[3. Основи на WSDL 6](#_Toc387153510)

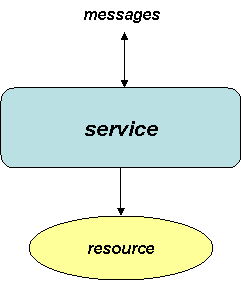
[4. История 6](#_Toc387153511)

[5. Структура 6](#_Toc387153512)

[6. Източници 6](#_Toc387153513)

1. Същност
   1. Дефиниция на W3C: WSDL представлява XML формат, който описва мрежови услуги като множество от крайни точки, които оперират чрез съобщения. Те съдържат или документно ориентирана или процедурно ориентирана информация. Операциите и съобщенията се описват абстрактно и тогава се връзват с конкретен мрежови протокол и формат на съобщението, за да се дефинира крайна точка. Свързаните конкретни крайни точки се комбинират в абстрактни крайни точки (услуги). WSDL е разширим, за да позволи описването на крайни точки и техните съобщения, без значение от формата на съобщението или мрежовите протоколи, които се използват за комуникация. Това въпреки, че единствените връзки, описани в този документ, описват как да използваме WSDL в съответствие със SOAP 1.1, HTTP GET/POST и MIME.
2. Бръз преглед на концепцията за уеб услуги

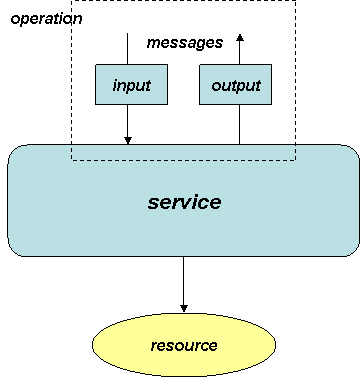
XML Web архитектура за услуги определя стандартен механизъм за предоставяне на ресурси чрез XML съобщения. Възможността да имат достъп до ресурсите чрез просто предаване на XML съобщения чрез стандартни протоколи като TCP, HTTP, SMTP значително улеснява летвата за потенциалните потребители. Терминът "Web услуга" (или просто "услуга") обикновено се отнася до част от кода, която прилага XML интерфейс към ресурсите, които в противен случай биха били трудно достъпни (вж. Фигура 1).



Тази архитектура позволява на всеки потребител с помощта на XML да се възползва от съответните Web услуги. Все пак, за да се постигне това , потребителите трябва да определят точния XML интерфейс , заедно с различни други детайли съобщение априори . XML схемата може частично да се запълни тази нужда, защото тя позволява на разработчиците да се описват структурата на XML съобщенията. XML схемата сама по себе си обаче, не може да се опише допълнителните подробности, участващи в общуването с уеб услуга.

Схемата просто казва как могат да се използват XML съобщенията, но не и как те си взаимодействат. Например, ако има един XML елемент с име Add и друг с име AddResponse, най-вероятно те ​​са свързани един с друг, но няма начин да се посочи в схемата .

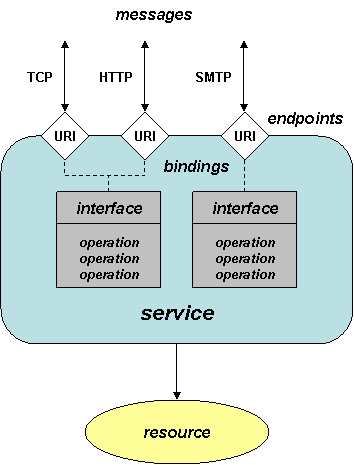
Обмена на съобщения се нарича и операция. Операциите са това, от което потребителите се интересуват най-много, тъй като те са пресечна точка на взаимодействие с услугата (вж. Фигура 2 ) . Всеки път, когато се обърне към нова уеб услуга , първо проверяваме списъка на поддържаните операции, за да се получи цялостно усещане за това, което предлага .



Обичайно е за разработчиците да групират подобни операции в интерфейси. Потребителите трябва да са наясно с тези групи , тъй като те влияят на начина, по който те пишат своя код . Това е особено важно за разработчиците , работещи с уеб услуги в обектно-ориентирани среди , тъй като XML интерфейсите могат да преобразуват програмните интерфейси (или абстрактни класове) в определен език по избор.

Потребителите трябва да знаят какъв протокол за комуникация да използват за изпращане на съобщения до услугата, заедно със специфичните механики, свързани с използването на даден протокол, като например използването на команди, хедъри и кодовете за грешки .

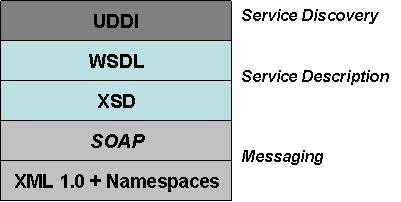
Услугите може да поддържа множество байндинги за даден интерфейс , но всеки от тях трябва да бъде достъпен в уникален адрес идентифициран с URI, или също наречен крайна точка (endpoint) (вж. Фигура 3 ) .



Потребителите трябва да открият всички детайли , описани по-горе , преди те да могат да си взаимодействат с уеб услуга. Web Services Description Language (WSDL) предоставя XML граматика за описване на тези подробности. WSDL допълва, там където XML Schema не мое да се справи чрез предоставяне на начин за групиране на съобщения в операции и операции в интерфейси. Това също осигурява начин за определяне и свързване на всеки интерфейс и комбинация от протокол заедно с endpoint . Пълна дефиниция на WSDL съдържа цялата информация, необходима, за да се позове на уеб услугата. Фирми , които искат да се направи лесно за другите да имат достъп до техните услуги следва да предоставят определения WSDL .

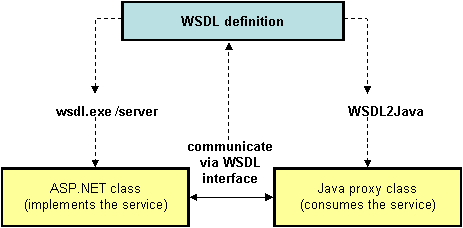
WSDL играе важна роля в цялостната архитектура на уеб услугите, тъй като той описва пълния договора за комуникация с дадено приложение (сходна с ролята на IDL в архитектурата DCOM) . Въпреки, че съществуват и други техники за описание на уеб услуги ,WS -I Basic профила Версия 1.0 препоръчва използването на WSDL и XML схема (вж. Фигура 4 ) за описание на уеб услуги . Това помага да се гарантира оперативната съвместимост на описателния слой на услугата.

.



Тъй WSDL е език четим за машините (т.к. това е просто един XML файл), инструменти и инфраструктура могат лесно да бъде изградена около него. Днес разработчиците могат да използват WSDL за да генерира код, който знае точно как да си взаимодействат с уеб услугата, който WSDL описва. Този вид подход крие досадните подробности, участващи в изпращане и получаване на SOAP съобщения по различни протоколи и прави уеб услуги, достъпни за широката публика.

Microsoft ®. NET платформата идва с помощна програма наречена wsdl.exe която генерира класове от WSDL дефиниции. Wsdl.exe може да генерира един клас за консумацията на услугата и друга за изпълнение на услугата. (Apache Axis идва с подобна програма, наречена WSDL2Java, която изпълнява същата функция за Java класове.) Класове, генерирани от същата WSDL дефиниция трябва да могат да комуникират един с друг чрез WSDL интерфейси, независимо от езиците за програмиране (виж Фигура 5).



WSDL 1.1 се счита за стандарта днес, защото се радва широка подкрепа и поддръжка. Повечето комплекти от инструменти за уеб услуги поддържат WSDL 1.1, но съществуват и някои проблеми със съвместимост в рамките на различните приложения. Много разработчици вярват, че обширната гъвкавостта на WSDL (и в резултат на сложността) е основният източник на тези проблеми. The WS-I е спомогнал да решат част от тези проблеми чрез насърчаване на разработчиците да използват някои части на спецификацията и не препоръчват използването на други.

1. Основи на WSDL

Основи

1. История
   1. WSDL 1.0 (септември 2000) е разработен от IBM, Microsoft и Ariba ,за да опишат своите инструменти SOAP уеб услуги. Реализацията е чрез комбиниране на два езика за описание: NASSL (Network Application Service спецификационен език) от IBM и SDL (Service Description Language) от Microsoft.
   2. WSDL 1.1, публикуван през март 2001 г., е формализирането на WSDL 1.0. Няма сериозни промени въведени между 1.0 и 1.1.
   3. WSDL 1.2 (юни 2003 г.) все още е работен проект на W3C. Според W3C: WSDL 1.2 е по-лесен и по-гъвкав от предишната версия. WSDL 1.2 прави опит за премахване на оперативно съвместими функции. WSDL 1.2 не е поддържан от голямата част от SOAP сървърни / доставчиците.
   4. WSDL 2.0 на W3C излиза юни 2007 година. WSDL 1.2 е преименувано на WSDL 2.0, тъй като има съществени различия от WSDL
2. Структура
3. Източници
   1. W3C Specification: <http://www.w3.org/TR/wsdl20/>
   2. Wikipedia (BG): <http://bg.wikipedia.org/wiki/WSDL>
   3. Wikipedia (EN): <http://en.wikipedia.org/wiki/Web_Services_Description_Language>
   4. MSDN - <http://msdn.microsoft.com>