

**Икономически университет - Варна Факултет
„Информатика“**



РЕФЕРАТ

на тема

**„Съвременни тенденции при уеб технологиите за пренос
на данни (Web Services, REST, SOAP)“**

Изготвил:

Божидар Тодоров Кираджийски

фак. № 120726

спец. ИКН, гр. 38

Варна

2021

СЪДЪРЖАНИЕ

I.Въведение	3
1.Какво представляват уеб услугите (Web Services) ?.....	3
II.REST и SOAP	4
1.Какво представлява REST ?.....	4
1.1.Клиент и сървър.....	5
1.2.Заявки и отговори.....	5
1.3.HTTP глаголи.....	6
2.Какво представлява SOAP ?.....	7
2.1.SOAP съобщения.....	8
2.2.SOAP характеристики.....	8
2.3.SOAP история.....	9
III.Разликите между REST и SOAP	10
1.1.Предимства на REST пред SOAP.....	10
1.2.Предимства на SOAP пред REST.....	11
IV.Заклучение	12

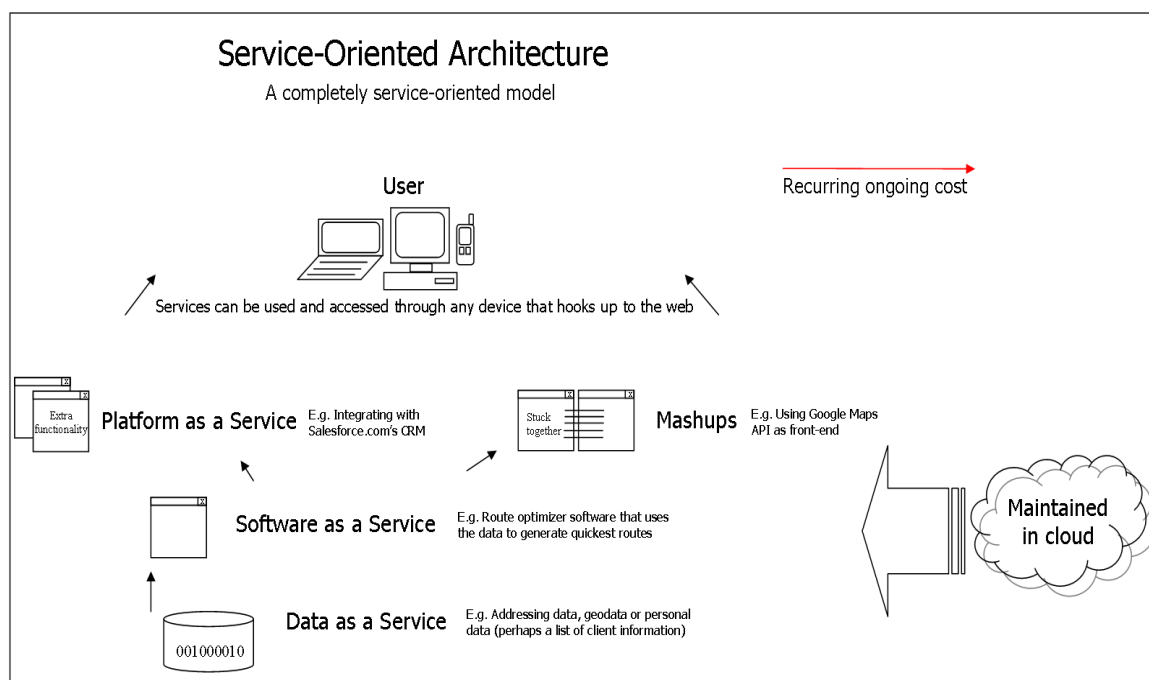
I. Въведение

1. Какво представляват уеб услугите (Web Services) ?

Историята на уеб услугите , започва от началото на 90-те. Уеб услугата представлява услуга , предлагана от електронно устройство на друго електронно устройство , комуникиращо помежду си чрез глобалната мрежа (World Wide Web) или сървър работещ на компютърно устройство , слушащ за заявки на определен порт през мрежа и обслужващ уеб документи (HTML , JSON , XML , изображения). Накратко уеб услугата е метод за комуникация между две електронни устройства през мрежа. Това е софтуерна функция, предоставена на мрежов адрес през мрежата с непрекъсната услуга. Уеб услугата позволява две софтуерни устройства да обменят информация чрез Интернет.

На практиката уеб услугата обикновено предоставя обектно-ориентиран уеб-базиран интерфейс на сървър на база данни, използван например от друг уеб сървър или от мобилно приложение, което предоставя потребителски интерфейс на крайния потребител.

Друго приложение, предлагано на крайния потребител може да бъде смесване, при което уеб сървърът консумира няколко уеб услуги на различни машини и компилира съдържанието в един потребителски интерфейс.



Целта на реферата е да информира читателите какво представляват :

- **Web Services**
- **REST**
- **SOAP**

Както и да сравни **REST** и **SOAP**.

II.REST и SOAP

1.Какво представлява REST ?

REST (Representational State Transfer) представлява архитектурен стил за реализация на уеб услуги между компютърните системи в мрежата, което улеснява системите да комуникират помежду си. Основната идея е да се определи системен ресурс, който се променя в резултат на взаимодействието между доставчика на услуги и потребителя. Архитектурният стил REST включва взаимодействията между сървър и клиент, осъществени по време на трансфера на данни.

1.1.Клиент и сървър.

В архитектурния стил REST, изпълнението на клиента и изпълнението на сървъра може да се осъществи независимо. Тоест ще се реализира дори без единия да знае за другия. Това означава че кодът от страна на клиента може да се променя по всяко време без да засяга работата на сървъра, а кодът от страна на сървъра може да се променя без да засяга работата на клиента.

1.2.Заявки и отговори.

В архитектурата на REST клиентите изпращат заявки за извличане или промяна на ресурси, а сървърите изпращат отговори на тези заявки.

Заявките обикновено се състоят от :

- HTTP глагол, който определя какъв вид операция да се извърши.
- Заглавна част (Header) , която позволява на клиента да предаде информация за заявката.
- Път към ресурса (Path to a resource).
- Незадължително тяло на съобщението, съдържащо данни.

Кодове на отговори:

Код на състояние	Значение
200 (OK)	Това е стандартният отговор за успешни HTTP заявки.
201 (CREATED)	Това е стандартният отговор за HTTP заявка , довело до успешно създаване на елемент.
204 (NO CONTENT)	Това е отговор за заявка, при която нищо не се връща в тялото на отговора.
400 (BAD REQUEST)	Заявката не може да бъде обработена, поради лош синтаксис на заявката.
404 (NOT FOUND)	Ресурсът не е намерен. Може да е бил изтрит или все още не е създаден.

Пример за заявка и отговор :

Ако искаме да разгледаме всички клиенти заявката ще изглежда така :

```
GET http://fashionboutique.com/customers
Accept: application/json
```

Отговорът е възможно да бъде :

```
Status Code: 200 (OK)
Content-type: application/json
```

1.3.HTTP глаголи.

Съществуват 4 основни HTTP глагола, които се използват за взаимодействие с ресурси в REST система :

- GET – извличане на определен ресурс (по ID) или колекция от ресурси.
- POST – създаване на нов ресурс.
- PUT – актуализиране на определен ресурс (по ID).
- DELETE – изтриване на определен ресурс (по ID).

2.Какво представлява SOAP ?

SOAP (Simple Object Access Protocol) е протокол за съобщения за обмен на информация между компютри. SOAP е приложение на спецификацията XML. SOAP осигурява лесна, разширяема и богата XML рамка за съобщения.

- SOAP е комуникационен протокол, предназначен за комуникация чрез Интернет.
- SOAP може да разшири HTTP за XML съобщения.
- SOAP осигурява транспорт на данни за уеб услуги.
- SOAP може да обменя пълни документи или да извиква отдалечена процедура.
- SOAP може да се използва за излъчване на съобщение.
- SOAP е независим от платформа и език.
- SOAP е XML начинът за определяне каква информация се изпраща и как.
- SOAP позволява на клиентските приложения лесно да се свързват с отдалечени услуги и да извикват отдалечени методи.

Въпреки че SOAP може да се използва в различни системи за съобщения и може да бъде доставен чрез различни транспортни протоколи, първоначалният фокус на SOAP е извикванията за отдалечени процедури, транспортирани чрез HTTP.

2.1.SOAP съобщения.

SOAP съобщението е обикновен XML документ, съдържащ следните документи :

ENVELOPE – Определя началото и края на съобщението. Това е задължителен елемент.

HEADER – Съдържа всички незадължителни атрибути на съобщението използвани при обработката на съобщението, или в междинна точка, или в крайната точка. Това е незадължителен елемент.

BODY – Съдържа XML данни, съдържащи изпращаното съобщение. Това е задължителен елемент.

FAULT – Този елемент предоставя информация за грешки, възникнали по време на обработката на съобщението. Това е незадължителен елемент.

2.2.SOAP характеристики.

SOAP има три основни характеристики :

- Разширяемост
- Неутралност
- Независимост (SOAP позволява всеки модел на програмиране)

Като пример за това какво могат да правят SOAP процедурите, приложението може да изпрати SOAP заявка до сървър, който има активирани уеб услуги – като база данни за цените на недвижимите имоти – с параметрите за търсене. След това сървърът връща SOAP отговор (XML- форматирани документи с получените данни), например цени, местоположение, характеристики. Тъй като генерираните данни идват в стандартизиран машинно разбираем формат, заявяващото приложение може да ги интегрира директно.

Следващият блок изобразява структура на SOAP съобщение :

```
<?xml version = "1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV = "http://www.w3.org/2001/12/soap-envelope"
  SOAP-ENV:encodingStyle = "http://www.w3.org/2001/12/soap-encoding">

  <SOAP-ENV:Header>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    ...
    <SOAP-ENV:Fault>
      ...
    </SOAP-ENV:Fault>
    ...
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.3.SOAP история.

SOAP е проектиран като протокол за достъп до обекти и е пуснат като XML-RPC през 1998 г. Спецификацията на SOAP се поддържа от работната група на XML на World Wide Web Consortium до затварянето на групата през 2009 г. След представянето на SOAP за пръв път, той се превръща в основен слой на по-сложен набор от уеб услуги.

III.Разликите между REST и SOAP

REST работи чрез единичен, последователен интерфейс за достъп до именувани ресурси. Най-често се използва когато излагате публичен API през интернет. SOAP от друга страна излага компонентите на логиката на приложенията като услуги, а не като данни. Освен това той работи чрез различни интерфейси. Казано по-просто, REST има достъп до данни, докато SOAP извършва операции чрез по-стандартизиран набор от модели за съобщения. И все пак в повечето случаи може да се използва, или REST, или SOAP, за да се постигне един и същ резултат, с някои разлики в начина на конфигурация.

1.1.Предимства на REST пред SOAP.

1. **REST** позволява по-голямо разнообразие от формати на данни, докато **SOAP** позволява само XML.
2. В комбинация с JSON (който обикновено работи по-добре с данни и предлага по-бързо анализиране), **REST** обикновено се счита за по-лесен за работа.
3. Благодарение на JSON, **REST** предлага по-добра поддръжка за клиентите на браузъра.
4. **REST** осигурява превъзходна производителност, особено чрез кеширане на информация, която не е променена и не е динамична.
5. Това е протоколът, който се използва най-често за основни услуги като: Yahoo, Ebay, Amazon, Google.

6. **REST** обикновено е по-бърз и използва по-малка честотна лента. Също така е по-лесно да се интегрира със съществуващите уебсайтове, без да е необходимо да се рефакторира инфраструктурата на сайта. Това дава възможност на разработчиците да работят по-бързо, вместо да отделят време за пренаписване на сайт от нулата. Вместо това те могат да добавят допълнителна функционалност.

1.2.Предимства на SOAP пред REST.

1. Стандартният HTTP протокол на SOAP го улеснява да работи през защитни стени и проксита без модификации на самия SOAP протокол.
2. Освен това, въпреки че е рядко необходимо, някои случаи на употреба изискват по-голяма надеждност на транзакциите от колкото това което може да се постигне с HTTP (което ограничава REST в това качество).
3. В някои случаи проектирането на SOAP услуги всъщност може да бъде по-малко сложно в сравнение с REST. За уеб услуги, които поддържат сложни операции, изискващи поддържане на съдържание и контекст. Проектирането на SOAP услуга изисква по-малко кодиране в приложния слой за транзакции, сигурност, довери и други елементи.

IV. Заключение

В крайна сметка най-добрият протокол е този, който има най-голям смисъл за организацията, типовете клиенти, които трябва да поддържате, и това, от което се нуждаете по отношение на гъвкавостта. Повечето нови API са изградени с помощта на REST и JSON, просто защото обикновено консумират по-малко честотна лента и са по-лесни за разбиране както за разработчици, прилагащи първоначални API, така и за други разработчици, които могат да напишат други услуги срещу него. Тъй като се използва по-лесно от повечето съвременни уеб браузъри, REST + JSON се превърна в дефакто технологията за повечето публични API. SOAP обаче остава ценен протокол при някои обстоятелства. Освен това не е нужно да търсите далеч, за да намерите упорити фенове, които се застъпват за SOAP за определени случаи на употреба.

Използвана литература

1. *Wikipedia – Web Services* -
https://en.wikipedia.org/wiki/Web_service
2. *TutorialsPoint*-
https://www.tutorialspoint.com/webservices/what_are_web_services.htm
3. *Уикипедиа - Веб услуги* -
https://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1_%D1%83%D1%81%D0%BB%D1%83%D0%B3%D0%B0
4. *Codecademy* -
<https://www.codecademy.com/articles/what-is-rest>
5. *Red Hat* -
<https://www.redhat.com/en/topics/api/what-is-a-rest-api>
6. *Wikipedia - REST* -
https://en.wikipedia.org/wiki/Representational_state_transfer
7. *Уикипедиа – REST* -
<https://bg.wikipedia.org/wiki/REST>
8. *TutorialsPoint* -
https://www.tutorialspoint.com/soap/soap_message_structure.htm
9. *Wikipedia - SOAP* -
<https://en.wikipedia.org/wiki/SOAP>

10. *Yukunedia* - SOAP -
<https://bg.wikipedia.org/wiki/SOAP>
11. *Guru99* - <https://www.guru99.com/soap-simple-object-access-protocol.html>
12. *stack overflow* -
https://stackoverflow.com/questions/19884295/soap-vs-rest-differences?fbclid=IwAR18QAJ-AiOga2NasVOnoxI3A_3-PdlZksCBp06W4vcmgF_FXJkb2J0HCgo
13. *Guru99* - <https://www.guru99.com/comparison-between-web-services.html?fbclid=IwAR0sjPkyHkV7H9br-CEE1CUz-ObAAAdPP7qdUO8Pxf4ZF2m8nzy0b0CRJIa8#:~:text=KEY%20DIFFERENCE-,SOAP%20stands%20for%20Simple%20Object%20Access%20Protocol,stands%20for%20Representational%20State%20Transfer.&text=SOAP%20needs%20more%20bandwidth%20for,%2C%20XML%2C%20HTML%20and%20JSON>
14. *Stackify* - <https://stackify.com/soap-vs-rest/>
15. *MuleSoft* -
<https://www.mulesoft.com/resources/api/what-is-rest-api-design>

