



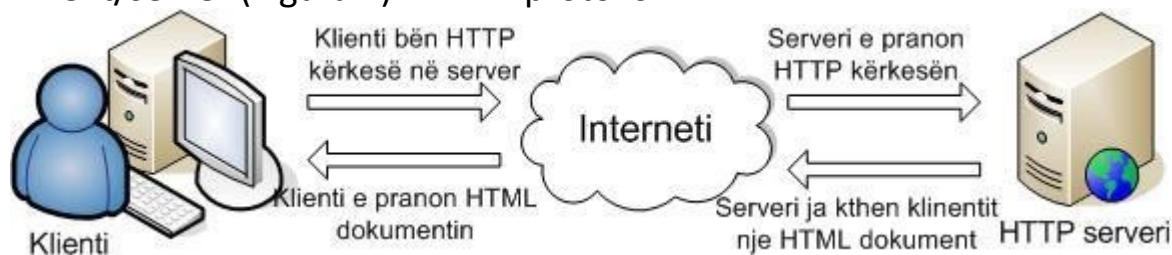
Programimi i distribuar

Pjesa 6 – Web shërbimet

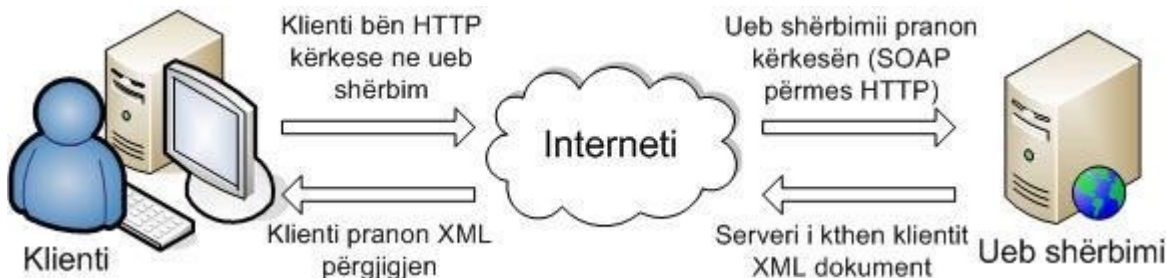
Prof. Asoc. Dr. Ermir Rogova

Çka janë web shërbimet?

- Web shërbimet janë një teknologji që mundësojnë qasjen e funksioneve në largësi përmes internetit
- Komunikimi përmes Web shërbimeve
- Komunikimi klient/server (Figura 1) – HTTP protokollit



- Komunikimi klient/web shërbim (Figura 2) – SOAP përmes HTTP protokollit
- Përgjigjet kthehen si XML dokumente
- HTTP është protokoll i nënkuptuar për web shërbimet, mirëpo nuk është protokoll i vetëm për komunikimin me web shërbime

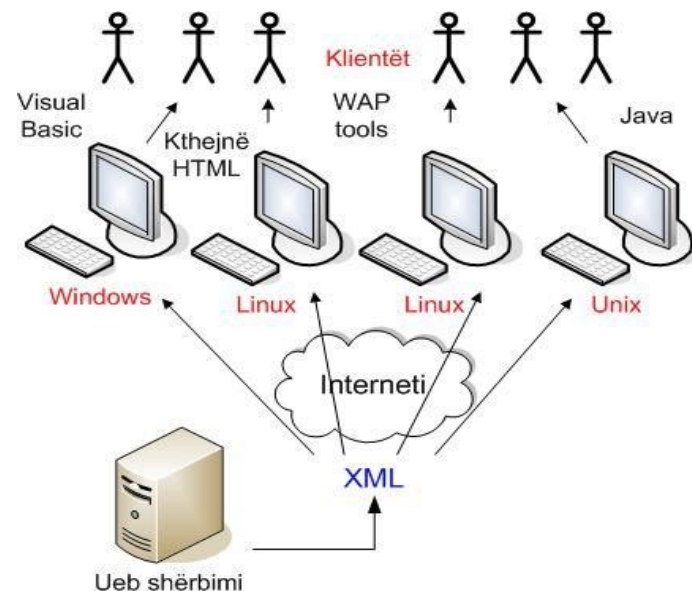


Arkitektura e web shërbimeve

- Web shërbimet ofrojnë aftësi të komunikimit ndërmjet sistemeve të ndryshme të aplikacioneve softuerike, të cilat ekzekutohen në platforma të ndryshme dhe në sisteme operative të ndryshme
- Nuk është e rëndësishme se ç'farë lloji i kompjuterit e dërgon SOAP mesazhin
- Nuk është e rëndësishme se ç'farë sistemi operativ përdorim
- Nuk është me rëndësi nga cili kënd i botës dërgohet mesazhi
- Nuk është me rëndësi gjuha programuese që ndërtohet web shërbimi

Komponentet kyç të web shërbimeve

- Komponentët kyç:
 - HTTP – Hypertext Transfer Protocol
 - XML – eXtensible Markup Language
 - SOAP – Simple Object Access Protocol
 - WSDL – Web Services Description Language
 - UDDI – Universal Discovery, Description Language
- HTTP - është një protokoll që është prezent kudo, që praktikisht ekzekutohet kudo në Internet.



XML

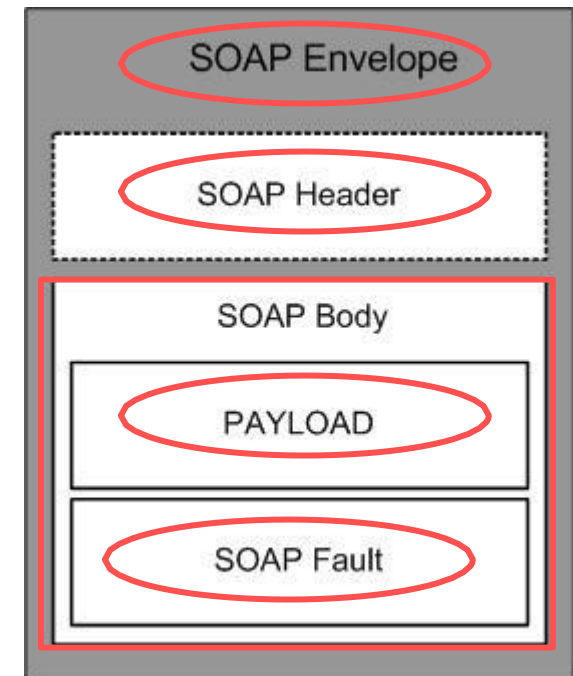
- Gjuha markuese XML përshkruan strukturën dhe kuptimin e dokumentit mirëpo nuk përshkruan formatizimin e elementeve në dokument
- Web shërbimet bazohen në gjuhën XML
 - XML është shkurtesë e EXtensible Markup Language
 - XML është gjuhë markuese ngjashëm me HTML
 - XML është dizajnuar për të përshkruar të dhënat
 - XML tagjet nuk janë te para definuara. Vetë duhet ti definojmë tagjet
 - XML përdor Document Type Definition (DTD) ose XML Schema për përshkrimin e të dhënave
- DTD i definon tagjet që janë të lejuara në një XML dokument dhe marrëdhëniet në mes tyre.
- XML Schema - XML Schema e bën përshkrimin e strukturës së XML dokumentit.
- XML namespaces – përdoren për te ju ikur konflikteve të emërtimeve të njëjtë të tagjeve

SOAP

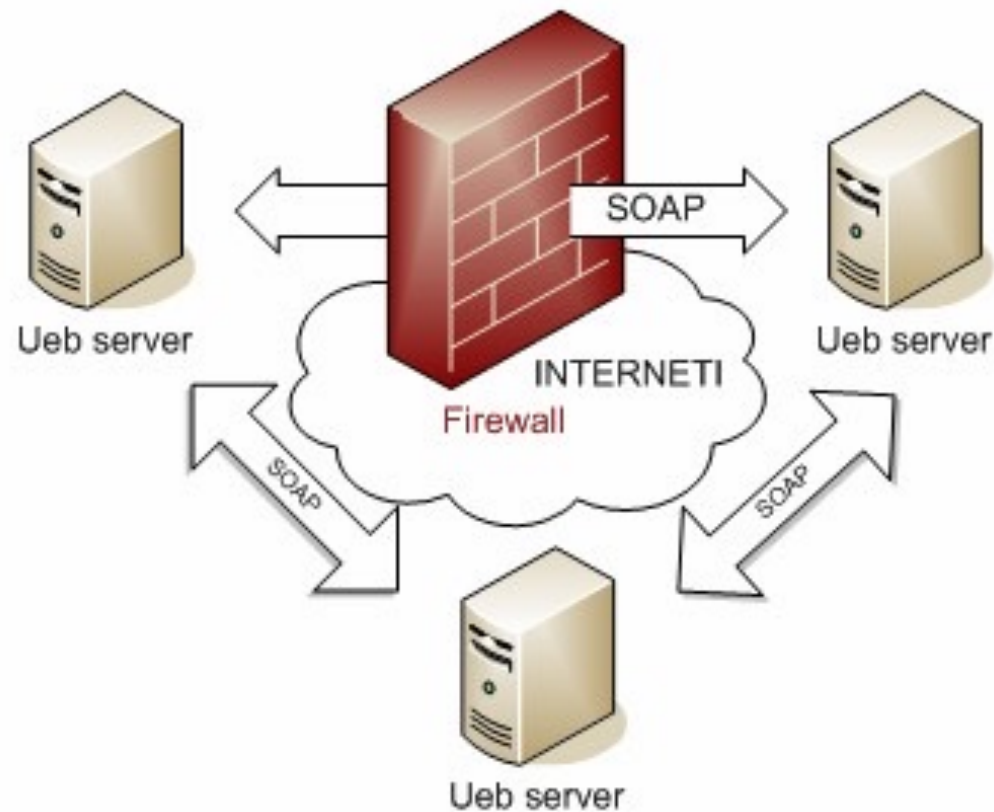
- SOAP është shkurtese e “Simple Object Access Protocol”
- SOAP është protokoll për komunikim që shërben për komunikim ndërmjet aplikacioneve
- SOAP është format për dërgimin e mesazheve që komunikon përmes Internetit
- SOAP është platforme e pavarur dhe gjuhe e pavarur
- SOAP bazohet ne XML
- SOAP lejon kalimin e mureve mbrojtës (firewall)

Struktura e SOAP mesazhit

- **SOAP Envelope** - SOAP Envelope është element i obligueshëm i cili paraqet elementin rrënjë të një SOAP mesazhi. Ky element e definon XML dokumentin si një SOAP mesazh.
- **SOAP Header** - SOAP heder nuk është element i obligueshëm për një SOAP mesazh. Ky element përmban informacionet e veçanta të aplikacionit (p.sh si autentifikimin, nënshkrime digjitale, pagesat etj.) rreth SOAP mesazhit
- **SOAP Body** - SOAP Body është element i detyrueshëm për një SOAP mesazh, ku në vete përmban mesazhet aktuale që janë vendimtare në fund të SOAP mesazhit **PAYLOAD** - Paraqet përmbajtjen e dokumentit që ne dëshirojmë ta dërgojmë.
- **SOAP Fault** – SOAP fault mesazhi është një mekanizëm që bën raportimin e gabimeve që ndodhin gjatë dërgimit të SOAP mesazhit.



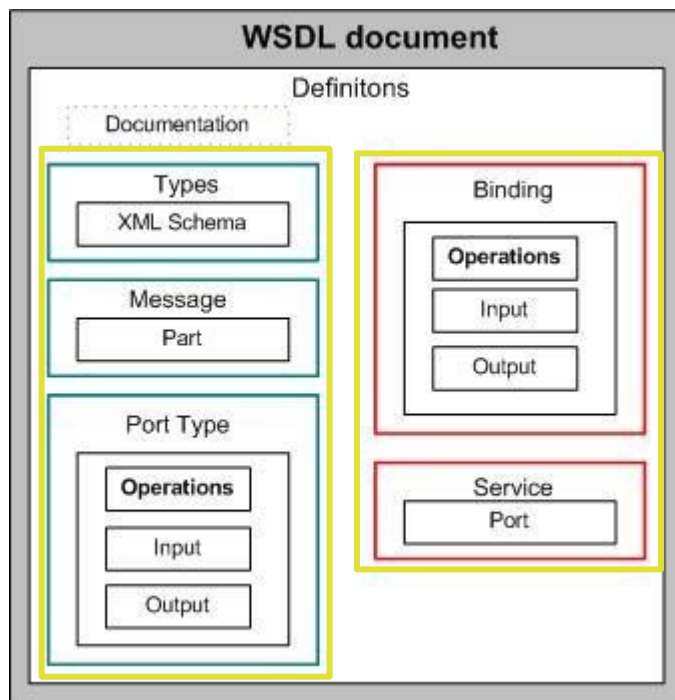
Depërtimi i SOAP mesazhit



WSDL

- WSDL është shkurtese për “Web Services Description Language”
- WSDL shkruhet në XML
- WSDL është XML dokument
- WSDL përdoret për të përshkruar web shërbimet
- WSDL poashtu përdoret për ti vendosur web shërbimet

Struktura e WSDL dokumentit



Abstrakte

- `<wsdl: types>`
- `<wsdl: message>`
- `<wsdl: operation>`
- `<wsdl: portType>`

Konkrete

- `<wsdl: binding>`
- `<wsdl: port>`
- `<wsdl: service>`

- **Types** - elementi types definon tipin e të dhënave që përdor ai web shërbim.
- **Message** - message përshkruan përmbajtjet e mesazheve duke përdorur XML skemat të ndërtuara
- **PortType** - elementi portType e definon një operacion si një bashkësi të mesazheve që fillon dhe mbaron me web shërbimin.
- **Binding** - elementi binding e definon një format të veçantë për secilin portType element në WSDL.
- **Service** - Ky seksion na tregon aktualisht adresat e ndryshme që mund të përdoren për të komunikuar me këtë web shërbim.

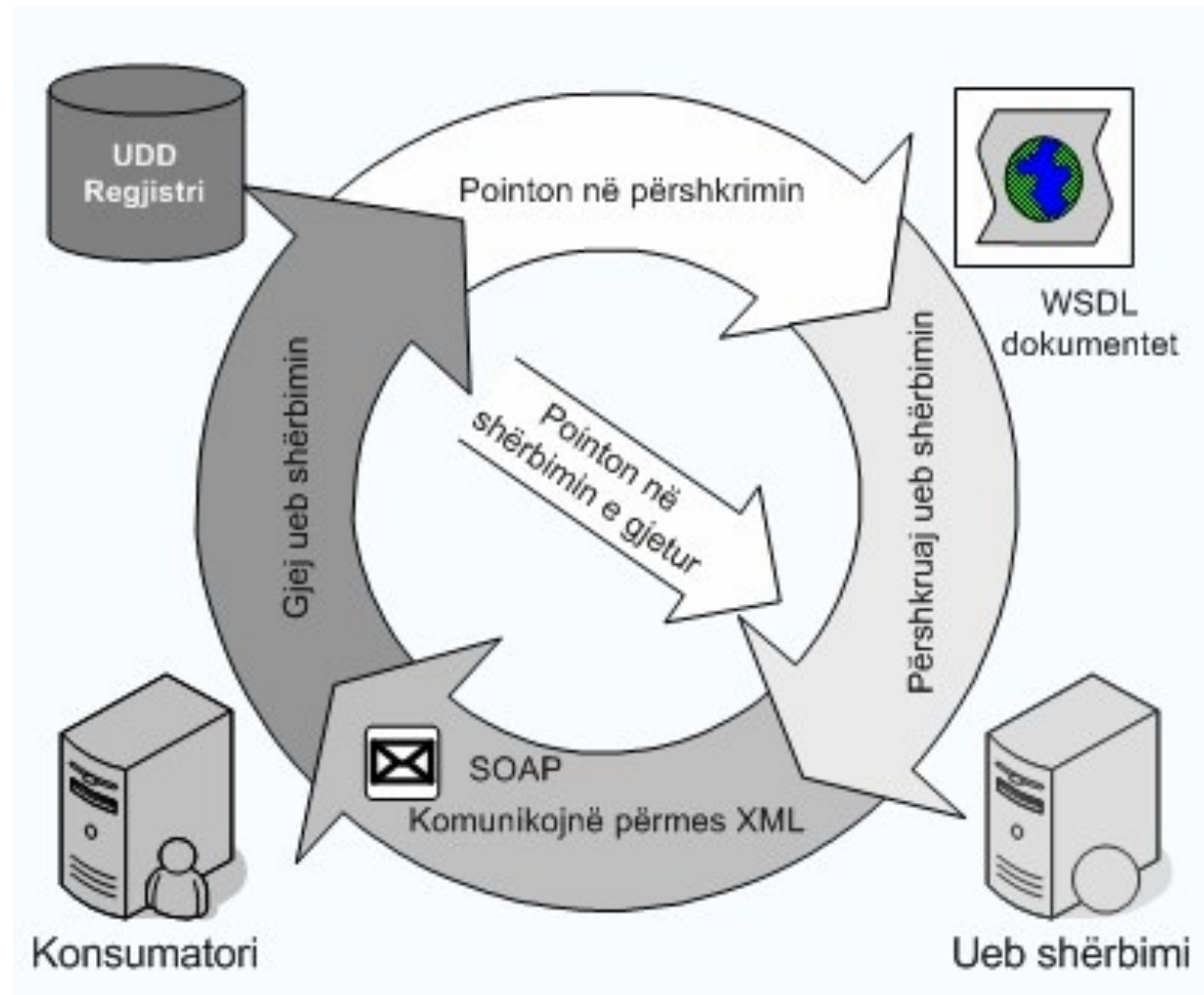
UDDI

- UDDI është shkurtesë për “Universal Description, Discovery and Integration”
- UDDI është platformë e pavarur, është regjistër i bazuar në standardin XML i cili shërben për ruajtjen e informacioneve të web shërbimeve
- Kjo ju shërben bizneseve në mbare botën ti listojnë web shërbimet e tyre dhe të kërkojnë shërbime tjera apo aplikacione softverike që bashkëveprojnë me internetin.
- UDDI komunikon përmes SOAP protokollit

Regjistrat e UDDI-së

- Regjistrat e UDDI-së mund të jenë të tipeve të ndryshme të cilat janë:
 - **Publik** - Ky regjistër është i hapur për kërkime publike. Të gjitha të hyrat në regjistrin publik kopjohen në të gjithë regjistrat e tjerë publik.
 - **Privat** - Ky regjistër qëndron prapa murit mbrojtës të kompanisë[PK03]. Qëllimi i këtij regjistri është kërkimi i web shërbimeve të brendshme. Këtij regjistri mund ti qasen p.sh. vetëm anëtarët e një ndërmarrje.
 - **Limituar** - Regjistrin të limituar mund ti qasen vetëm disa kompani të caktuara që ju është lejuar e drejta për t’ju qasur këtyre regjistrave.
- Informacioni në një regjistër është krijuar nga tri tipe, të cilat janë:
 - **Faqet e bardha** - Faqet e bardha përmbajnë informacion bazike për kontaktim. Mund të përmbajnë adresat, emrat kontaktues dhe numra të telefonit.
 - **Faqet e verdha** - Faqet e verdha shërbejnë për shfrytëzuesit që dëshirojnë të kërkojnë dhe të dinë për tipet e shërbimeve biznesore që kompania i ofron.
 - **Faqet e gjelbërta** - Faqet e gjelbërta të japin informacione teknike për web shërbimet që janë të ekspozuara nga bizneset e ndryshme. p.sh Përshkrimin e shërbimit, rregullat e biznesit etj.

Ndërlidhja



XML vs HTML

- XML pershkruan strukturen por jo formatin

- HTTP kodi

```
<td>Punëtori</td>  
<td>Bledar</td>  
<td>Hisa</td>  
<td>07091984</td>  
<td>Prishtine </td>  
<td>1450000329990</td>
```

- XML kodi

```
<Punëtori>  
<Emri>Bledar</Emri>  
<Mbiemri>Hisa</Mbiemri>  
<ID>07091984</ID>  
<Vendi>Prishtine</Vendi>  
<IDBankës>1450000329990</IDBankës>  
</Punëtori>
```

Përparësitë e përdorimit të web shërbimeve

- Përdorin platformë të pavarur
 - (Windows, Linux, Unix etj.)
- Sigurojnë kanal komunikimi në të njëjtën kohë
 - (Interneti)
- Mundësi komunikimi me ndërmarrje të ndryshme
 - (B2B dhe B2C)
- Funksionalitet në përdorim
- Gjuhë programuese të pavarur
 - (C#, C++, Visual Basic etj.)
- Komunikim të sigurtë
 - (web shërbimet për nga siguria janë në nivel të njëjtë me web aplikacionet)

REST

- REST është shkurtese për “Representational State Transfer”
- Ajo përfaqëson gjendjen e bazës së të dhënave në një kohë të caktuar. Por si ???
- REST është një stil arkitektonik i cili bazohet në standardet e internetit dhe protokollit HTTP.
- Në një arkitekturë të bazuar në REST, gjithçka është një burim (resurs).
- Një resurs arrihet përmes një ndërfaqeje të përbashkët bazuar në metodat standarde HTTP.
- Zakonisht kemi REST serverin i cili siguron qasje në burime dhe REST klientin i cili i çaset dhe modifikon burimet REST.

REST

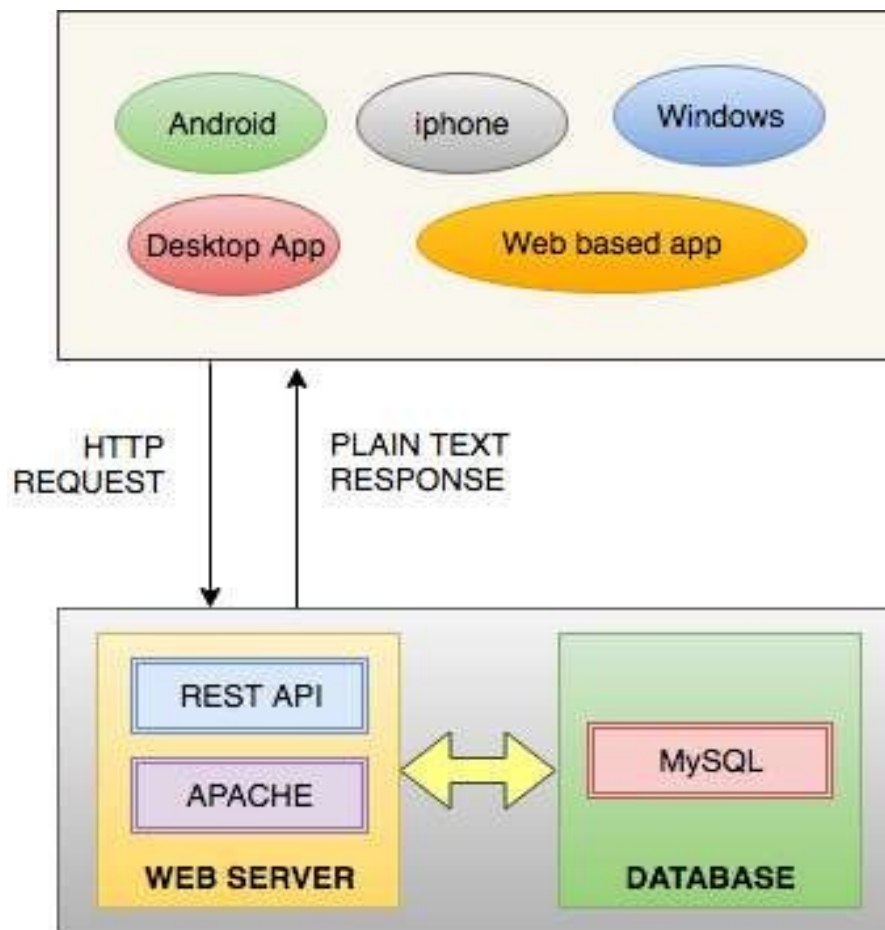
- Çdo resurs duhet të mbështesë operacionet e përbashkëta HTTP.
- Burimet identifikohen nga ID globale (të cilat zakonisht janë URI ose URL).
- REST lejon që burimet të kenë paraqitje të ndryshme, p.sh., tekst, XML, JSON, etj.
- Nuk ka gjendje. E shkëlqyeshme për sistemet e shpërndara.
- Komponentët pa gjendje mund të rishpërndahen lirisht nëse diçka dështon, dhe ato mund të shkallëzohen për të akomoduar ndryshimet e ngarkesës.
- Kjo sepse çdo kërkesë (request) mund të drejtohet në çdo instancë të një komponenti

Metodat HTTP

- Metodat PUT, GET, POST dhe DELETE përdoren zakonisht në arkitekturat e bazuara në REST
- Tabela e mëposhtme jep një shpjegim të këtyre operacioneve:

HTTP Method	CRUD Operation	Description
POST	INSERT	Addes to an existing resource
PUT	UPDATE	Overrides existing resource
GET	SELECT	Fetches a resource. The resource is never changed via a GET request
DELETE	DELETE	Deletes a resource

Arkitektura



Shembull i kërkesës HTTP

```
GET /doc/test.html HTTP/1.1
```

```
Host: www.test101.com
```

```
Accept: image/gif, image/jpeg, */*
```

```
Accept-Language: en-us
```

```
Accept-Encoding: gzip, deflate
```

```
User-Agent: Mozilla/4.0
```

```
Content-Length: 35
```

```
bookId=12345&author=Tan+Ah+Teck
```

Request Line

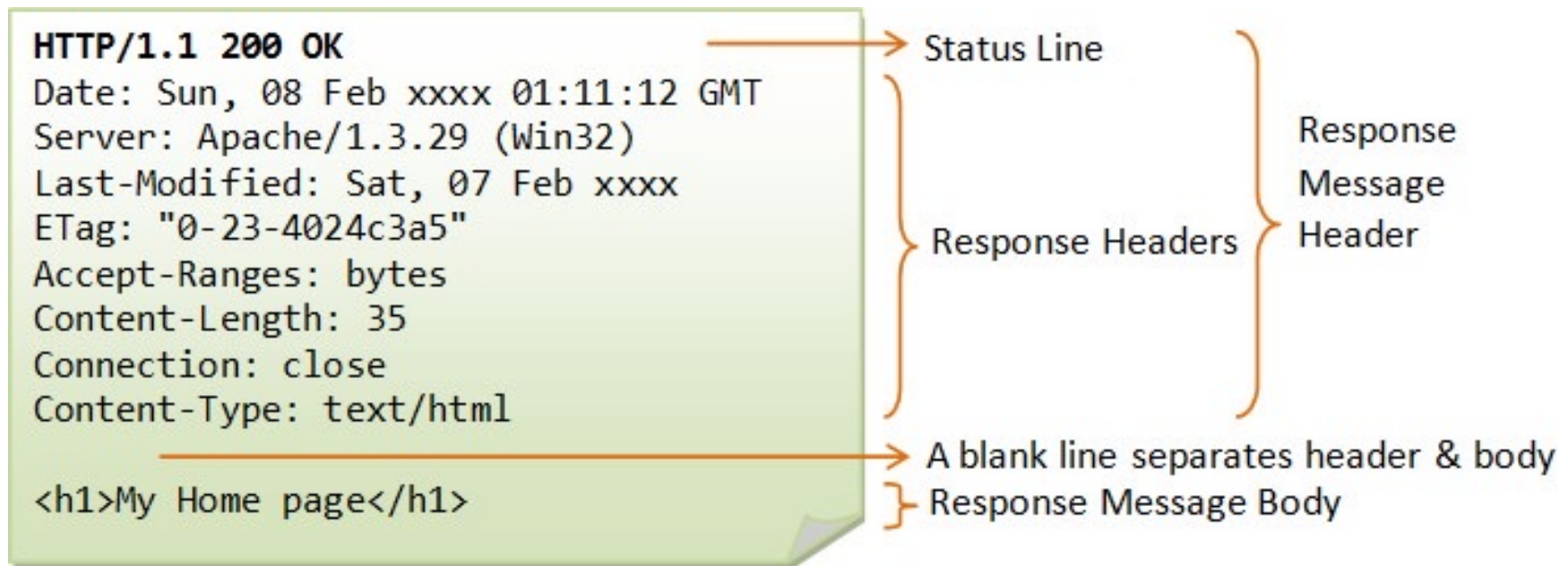
Request Headers

Request
Message
Header

A blank line separates header & body

Request Message Body

Shembull i përgjigjes HTTP



Lexo, Të gjithë qytetet për përdoruesin ID e së cilit është 1

HTTP REST API Request

HTTP REST Response:

```
HTTP/1.1 200 OK (285ms)
Date: Fri, 21 Apr 2017 10:27:20 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.1 e-fips PHP/7.0.16
X-Powered-By: PHP/7.0.16
Content-Length: 109
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json; charset=UTF-8
```

```
-----
{"status":"success","message":"City
List","data":[{"city_name":"Prishtina"}, {"city_name":"Prizreni"}]}
```

HTTP REST API Response

Kodet e statutit në përgjigjen HTTP

1xx	Informational Codes
2xx	Successful Codes
3xx	Redirection Codes
4xx	Client Error Code
5xx	Server Error Codes

Lista e kodeve : https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

Udhëzime për REST API

- Përdorni emra e jo folje në shteg/URI

Purpose	Method	Incorrect	Correct
Retrieves a list of users	GET	/getAllCars	/users
Create a new user	POST	/createUser	/users
Delete a user	DELETE	/deleteUser	/users/10
Get balance of user	GET	/getUserBalance	/users/11/balance

Udhëzime për REST API

- Përdorni vetëm emrat shumës për të gjitha burimet.

/veturat në vend të /vetura

/përdoruesit në vend të /përdoruesi

/produktet në vend të /produkti

/settings në vend të /setting

Udhëzime për REST API

- Metoda GET nuk duhet të ndryshojë gjendjen
 - Përdor metodat PUT, POST dhe DELETE në vend të metodës GET për të ndryshuar gjendjen

GET /përdoruesit/711?Aktivizoni

ose

GET /përdoruesit/711/aktivizo

Udhëzime për REST API

- Përdorni nën-burimet për lidhje:
- Nëse një burim është i lidhur me një burim tjetër përdorni nënburimet

GET /veturat/711/shoferët/ (Kthen listën e shoferëve të veturave 711)

GET /veturat/711/shoferët/4 (Kthen shoferin #4 për veturën 711)

Udhëzime për REST API

- Përdorni HTTP headerët për formatet e serializimit:
 - Të dy, klienti dhe serveri duhet të dinë se cili format përdoret për komunikimin.
 - Formati duhet të specifikohet në HTTP-Header.
 - *Content-Type* përcakton formatin e kërkesës.
 - *Accept* përcakton një listë të formateve të pranueshme të përgjigjeve.

Udhëzime për REST API

- Versionimi është i rëndësishëm:
 - Bëni të detyrueshëm versionin API dhe mos lëshoni një API të paversionuar
 - Përdorni një numër rendor të thjeshtë dhe shmangni shënimin e pikave siç është 2.5
 - Në shembull përdorer url për versionin API duke filluar me shkronjën "v"
/blog/api/v1

Udhëzime për REST API

- Trajto gabimet duke përdorur status codet e HTTP

200	OK (Everything is working)	403	Forbidden (The server understood the request, but is refusing it or the access is not allowed)
201	OK (New resource has been created)	404	Not found (There is no resource behind the URI)
204	OK (Resource successfully deleted)	405	Method not allowed
400	Bad Request (The request was invalid or cannot be served. The exact error should be explained in the error payload. E.g. „The JSON is not valid“)	408	Request timeout
401	Unauthorized request requires an authenticated user (The user authentication)	500	Internal server error

Udhëzime për REST API

- Filtrimi:
 - Përdorni një parametër unik të pyetjes për të gjitha fushat ose një gjuhë pyetjeje për filtrimin.

GET /veturat?ngjyra=kuq (Kthen listën e makinave të kuqe)

GET /përdoruesit?emri=Ermir (Kthen listën e përdoruesve emri i të cilëve është Ermir)

Udhëzime për REST API

- Sortimi (rënditja):
 - Lejoni rënditjen ngjitës dhe zbritës mbi shumë fusha.

GET /veturat?rendit=-prodhuesi, +modeli

- Kjo kthen një listë të veturave të renditura nga prodhuesit në rënie ($Z \rightarrow A$) dhe modeli në ngjitje ($A \rightarrow Z$)

Udhëzime për REST API

- Faqosja:
 - Përdorni **limit** dhe **offset**. Është fleksibile për përdoruesit dhe i zakonshëm në bazat e të dhënave.
 - Parazgjedhja duhet të jetë limit=20 dhe offset=0

GET /veturat?offset=10&limit=5

Udhëzime për REST API

- Pseudonimet (aliases) për pyetjet e shpeshta:
 - Për ta bërë përvojën e API-së më të këndshme për konsumatorin e zakonshëm, merrni parasysh paketimin e grupeve të kushteve në shtigje REST lehtësisht të arritshme.
 - Për shembull, merrni parasysh rastin e mëposhtëm të përdorimit:

POST /përdoruesit (Krijon një përdorues të ri)

POST /përdoruesit/hyrja (Krijon token auth për vërtetimin e përdoruesit)

Udhëzime për REST API

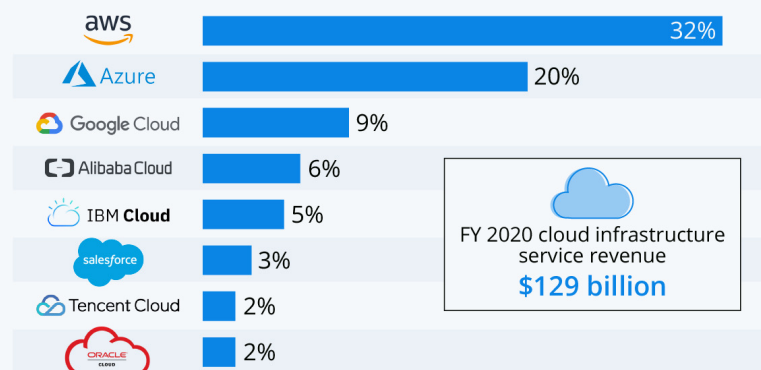
- Për zhvillim të API-të REST me lehtësi
 - Slim – microframework për php
- www.slimframework.com

Tregu i web shërbimeve

- Web shërbimet kanë filluar të përdoren në vitin 2000 me shpresa shumë të mëdha
- Web shërbimet në përgjithësi llogariten si “hapi i ardhshëm” në zhvillimin e integritit të aplikacioneve dhe të ndërveprimit të tyre
- Në vitin 2020 web shërbimet kanë arritur vlerën monetare prej \$130 billion

Amazon Leads \$130-Billion Cloud Market

Worldwide market share of leading cloud infrastructure service providers in Q4 2020*



FY 2020 cloud infrastructure service revenue
\$129 billion

* includes platform as a service (PaaS) and infrastructure as a service (IaaS) as well as hosted private cloud services

Source: Synergy Research Group



statista



Pyetje???