HTTP

HTTP basics

WWW-aplikacije i servisi koji se temelje na HTTP protokolu
HTTP protokol-klijent server arhitektura , asimetričan zahtijev-odgovor protokol gdje klijent(web browser ,deskop app,mobile app) izvlači informacije iz servera

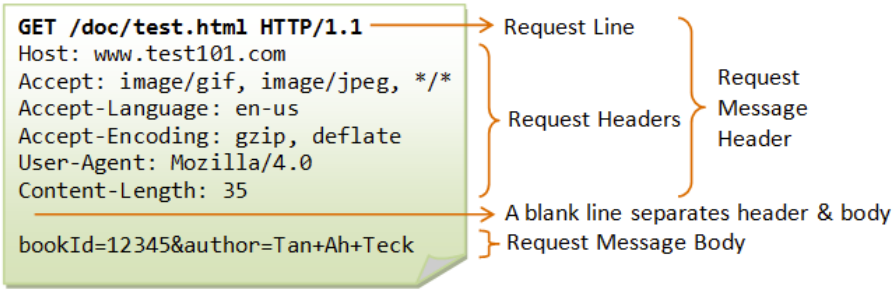
http :// www.fer.unzig.hr /zavod/zemris
protokol host reosurce path

port: 80-HTTP, 443-HTTPS ,21-FTP ,23-TELNET

URI(uniform resource identifier)-string koji se koristi da bi jedinstveno identificirao resurs preko web-a
protocol://hostname:port/path-and-file-name?parameters
protocol-protokol na aplikacijskoj razini kojeg koriste klijent i server
hostname- DNS ime domene ili IP adresa servera
port-TCP port kojeg server sluša za ulazni zahtijev od klijenta
path-and-file-name-ime i lokacija zatraženog resursa u serverskom baznom direktoriju
parameters- opcionalno,dodatno opisuje resurs

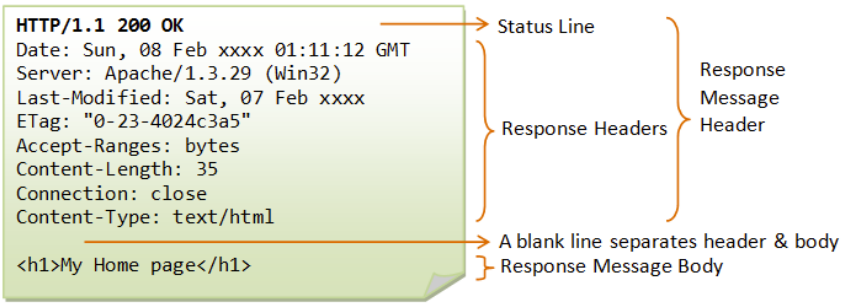
HTTP klijentski algoritam
1.Korisnik upisuje URI web stranice 2.browser pita DNS za IP 3. DNS daje IP 4. Browser otvara konekciju prema IP adresi i TCP portu 5.browser šalje HTTP zahtijev web serveru 6.server mapira URI u lokalnu datoteku 7. Server vraća HTTP odgovor 8. Browser formira odgovor ,i radi GUI i prikazuje web stranicu

HTTP request message:



Request line
Request method name(GET) – informira server za operaciju koja se izvodi preko resursa
Request URI – specificira resurs na web serveru preko kojeg server bi trebao izvoditi request operaciju
HTTP version –klijent specificira verziju HTTP protokola
CRLF-new line
Request header
Request header name:request header value1 ,request header value2,...,CRLF
Request message body
Slobodan format ,proizvoljana duljina

HTTP response message:



Status line
HTTP verzija:server specificira verziju HTTP protokola koji se koristi u odgovoru.Verzija mora biti jednaka ili niža vrijednosti klijentovog zahtijeva
Status code : broj generiran od strane servera koji označava odgovor zahtijeva
Reason phrase:
CRLF

Response header
Response header name:resp header value 1,resp header value 2,... CRLF
Response message body

REST ograničenje

Mrežna ograničenje

Klijent –server

Klijent traži neke resurse a server mu ih daje.Dvije elementarne komponente svakog web sustava

Slojevit sustav

Slojeviti sustav koji omogućuje da mrežni posrednici budu tranparentno postavljeni između klijenta i servera(proxy ,gateway),to je opcionalna komponenta web sustava.Prednost slojevitih sustava su (caching , load balancing)->skalabilnost, sigurnost,konverzija protokola
Proxy:spaja 2 ili više aplikacija koje imaju isti protokol.Daju sigurnost,(caching,load balancing) ->skalabilnost. Proxy sijedi između klijenta i servera,uzima sve HTTP zahtijeve i predaje zahtijeve serveru i može modificirati zahtijev ,te prima odgovore od servera i šalje ih natrag klijentu te isto tako može modificirati odgovor.
Sigurnost:filtrira porno sadržaj,detektira viruse
Caching:uzima objekte od orginalnog servera i sprema ih kod sebe za potrebe klijenata .
Load balancing:ili surogat,server acelerator,reverse proxy koji ujednačava zahtijeve između više slobodnih servera i tako poboljšava izvedbu servera tijekom heavy load.
Gateway:spaja 2 ili više aplikacija koji imaju različite protokole.

Sadržajna ograničenje

Adressability(adresibilnost)

Resursi su dobro definirana semantika koja ističe neki resurs od nekog drugog.Osnovni građevni blokovi ROA arhitekture.
Identifikator resursa:svaki resurs je globalno jedinstven,korišten da bi referencirao resurs sa udaljenog klijenta.

Reprezentacija resursa: podatkovni format korišten za reprezentaciju stanja resursa i za razmjenu stanja između klijenta i servera.Svi resursi reprezentiraju isto stanje.
URI: URL(uniform resource locator je najčešća forma URI koja opisuje specifičnu lokaciju resursa na pojedinom serveru i govori točno kako dohvatit resurs od fiksne lokacije),
URN(uniform resoruce name je jedinstveno ime za pojedini resurs koji dopušta resursu da se miče od mjesta do mjesta i da se pristupi na njega sa više protokola)

URL sintaxa :<scheme>://<user>:<pass>@<host>:<port>/<path>;<params>?<query>#<fragment>
Shema :koji protokol se koristi
User pass :neki servisi zahtijevaju korisničko ime i šifru
Host port: IP i port(80 za http)
Path:podskupine
Params: ulazni parametri,rijetko se koristi
Query: neka vrsta select upita
Fragment:npr ako želimo otvoriti dio nekog teksta „test.html#podskup“

Resursi+URI=>addresability :fina granulacija web servisa ,daje strukturu web servisa

Conectedness(povezanost)
ROA (representation state transfer) je stil arhitekture i prog paradigme za dizaj i razvoj softwera u obliku resursa sa RESTful sučeljima. Resursi mogu imati više reprezentacija(google com,google hr)
Hypermedia: je ne linearni sustav podataka koji omogućuje kompilaciju sadržaja multimedije na povezan način(tekst,video,zvuk,grafika).
HTML->ljudski web ,XML ,JSON->programmable web
HTML
XML: formati nastali od XML su XHTML,SOAP,RSS,Atom,OpenDocument
JSON:jezik za reprezentaciju jednostavne podatkovne stukture i asocijativnih polja koji se zovu objekti(objekti,polja,brojevi,string,boolean,null)
Media tip:HTTP se pouzda na media tipovima ,ne ekstenizijama podataka. Za raspoznati reprezentaciju se u request/response header stavlja „Content-Type :text/html, content –length:200 charset=us-ascii“

HTML	XML, JSON
Human web	Programmable web
Predefined and standardized markup semantics	Application-specific markup semantics

XML	JSON
Bulky representations due to extensive markup	Compact representations
Lack of data types	Native mapping to almost any programming language's data type system
Generic and unconstrained	Somewhat limited with a predefined set of data types

Interakcijska ograničenje
Uniform interfaces(ujednačeno sučelje)
Resource interface:svaki resurs ima svoj skup operacija (Command transfer-RPC)

Uniform interface:različiti resursi,isti i konačan skup operacija.Zbog njega svaki klijent može pristupiti svaki RESTful web servis na sličan način koristeći skup interakcijsih pravila(univerzalan klijent) (state transfer -RESTful)
CRUD :create(put,post) ,read(get,head,options),update(put),delete(delete)
Get:N-message body –dobiva resurs sa servera.URI zahtijev može se odnositi na statični resurs koji je spremljen u datoteci,statični ili dinamični resurs koji je spremljen u bazi,podatkovni proces(server izvrši proces povezan s danim URI).

Put-Y-stavlja podatke specificirane u tijelu zahtijeva prema serveru kao nova resursna reprezentacija.Put se koristi za kreiranje novog resursa(pošalje se zahtijev s nepostojećim URI,server napravi novi resurs i dodjeli to tom nepostoječem URI-u,server odgovori s „201 Created“) ili ažiriranja postojećeg resursa(server zamjeni postojeći resurs sa novim resursem i odgovori s „200 OK“).
PUT/product.txt HTTP/1.1 ->HTTP/1.1 201 Created Location:http:/...com/product.txt

Post-Y-šalje podatke prema serveru za procesiranje.Izradi podređeni resurs unutar kolekcije.Server odgovori sa „201 Created“. Koristi se za slanje e-mail,SMS,alert itd. Može se koristit i za tuneliranje ostalih CRUD metoda kada klijent ih ne podržava ali bi se takav način trebao izbjegavati.U praksi se koriste za podršku HTML formi u browserima(GET,POST).

Korištenje GET umjesto POST za slanje HTML.Prednost POST u odnosu na GET je da serveri često ograničavaju duljinu URL. Prednost GET u odnosu na POST je da je zahtijev zabilježen na klijentu.

Delete-N-uklanja resurse sa servera.
DELETE/prod.txt HTTP/1.1 ->HTTP/1.1 200 OK(ako je brisanje trenutno obavljeno) ili 202 Acepted(ako je duži period za obrisati)

Head-N-dobiva samo header resursa sa servera ali ne i reprezentaciju.Ponaša se kao GET ,ali server vraća samo head. Može se dobiti inforamcije o resursu bez da ga se uzme,saznati dal postoji,testirati dal je bio modificiran gledajući headers.

Options-N-određuje koja metoda je slobodna za izvedbu preko resursa
OPTIONS * HTTP/1.1 ->HTTP/1.1 200 OK Allow:GET,POST,PUT,OPTIONS
URI kao *

Status codes :numeric value + reason phrase

Overall range	Defined range	Category
100-199	100-101	Informational
200-299	200-206	Successful
300-399	300-305	Redirection
400-499	400-415	Client error
500-599	500-505	Server error

Metoda se gleda dal je sigurna(ako ima isti efekt na resurs kao da nije izvršena uopće) i idempotentna(ako višestruko izvršavanje preko istih resursa ima isti učinak kao i pojedino izvršavanje).
Metode :GET(YY),PUT(NY),POST(NN),DELETE(NY),HEAD(YY),OPTIONS(YY)

REST : URI(http://api.example.com),Metode(GET,PUT,POST,DELETE,HEAD,OPTIONS),Media tipovi(HTML,XML,JSON)

Statelessnes

Statles protokol(IP,HTTP) je protokol koji tretira svaki zahtijev kao nezavisna transakcija koja je nepovezana na bilo koji prijašnji zahtijev tako da komunikacija sadrži nezavisne parove zahtijeva i odgovora.Statles protokol ne zahtijeva od servera da vrati informacije o sesiji . Protokoli koji zahtijevaju čuvanje sesija su stateful protokoli.

Sa statelesnes ograničenjima prednosti su:

-Sesion state se čuva na klijentu (aplication state),dok server održava samo resource state

-skalabilnost je poboljšana zbog toga što se nemoraju alocirati resursi za čuvanje aplication state.

-vidljivost je poboljašna budući da sustav za praćenje nemora gledati dalje od pojedinog upita.

-pouzdanost je poboljšana zbog lakšeg oporavka od dijelomičnog pada.

Ograničenja su:

Ograničena izvedba mreže i ograničena kontrola servera za konzistenciju aplication state.

Prednosti za server: distribucija stateless aplikacija preko load balanced servera i caching.

Prednosti za klijenta:URI koji radi isto i na početku HTTP sesije ,i satima duge sesije

Cache

Web cache je mehanziam za privremeno spremanje web dokumenata kao HTML stranica radi reduciranja bandwith usage,server load ,smanjuje distance delays. Web cache kopira dokumente koje prolaze kroz caching proxy.

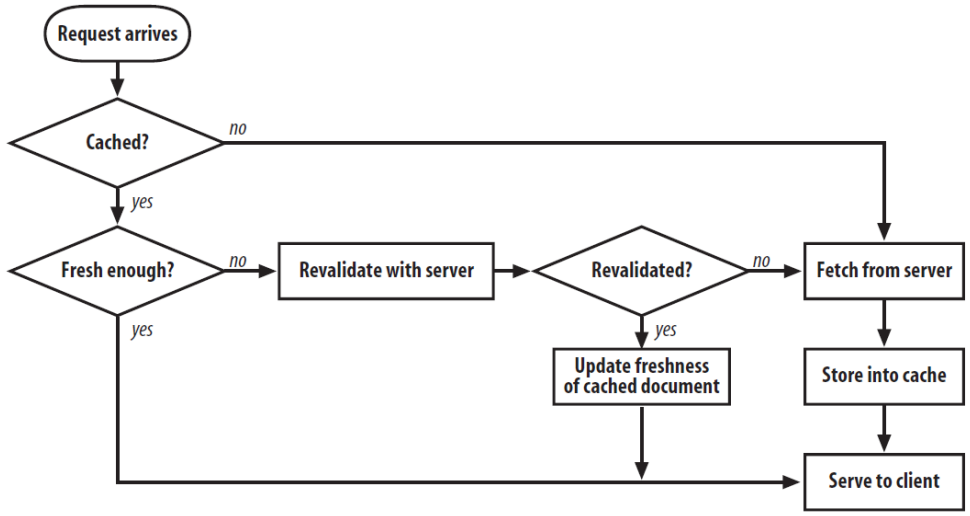
Cache miss:kod prvog zahtijeva za resuresem,resurs koji je dohvaćen sa orginalnog servera je spremljen u lokalnom cache.

Cache hit:kad se povno pojavi zahtijev za istim resursem,resurs je dohvačen sa lokalnog cachea umjesto orginalnog servera.

Cache revalidation: cache pita orginalni server dal je resurs promjenjen tj. revalidate hit /revalidate miss,provjera dal je cached resurs još uvjek sviježi(HTTP definira poseban upit na temelju kojeg se određuje sviježina->resoruce expiration)

Resoruce expiration:Expires(datum isteka) ,Cache control :max age(sekunde do isteka) ,Etag(version identifier)->nemora značiti da je resurs promjenjen ako je isteklo vrijeme/datum

Postavlja se kondicionalni GET zahtijev(GET/nest.html HTTP/1.0) a odgovor je (HTTP/1.0 304 Not Modified) ako nije promjenjen ,ako je promjenjen onda je (HTTP/1.0 200 OK) sa punim sadržajem resursa.Ako je resurs izbrisan onda je (HTTP/1.0 404 Not Found HTTP/1.0)



Hit rate :cache hits/number of request

Byte hit rate: broj byte dostavljenih iz cache/cijela veličinia svih odgovora

Privatni cache :je osobni cache za pojedinog korisnika te za njega nije potrebno puno memorije(web browseri imaju privatni cache ugrađen)
Javni cache : dijeli se unutar pojedinih mreža koje imaju caching proxy server

Code on demand

To je opcionalno ograničenje u REST-u.Klijentska funkcionalnost se može proširiti skidanjem i izvršavanjem koda u obliku apleta ili skripte(javascript skripta se pokreće kad se izabere HTML stranica sa skriptom u stranici). Korištenje code on demand **smanjuje vidljivo**st te zato je to ograničenje opcionalno.
Spriječeva pregled i filtiranje interakcije na mrežno vatrozidu te otvara mogučnost spremanja malware.
Kod caching ,kod se može spremiti ali rezultat izvršenja se nemože spremiti.