BOLUM 3

HTTP MESAJLARI

Eger HTTP Internetin kuryesisye,HTTP mesajlari etrafta dagitilan paketler diyebiliriz.Bolum 1 de,HTTP programlarinin birbirine mesaj yollayarak nasil islerini hallettigini gostermistik.Bu bolumde HTTP mesajlari hakkinda her seyden bahsedecegiz.Onlarin nasil yaratildigini ve onlari nasil anladigimizdan bahsedilecek.Bu bolumu okuduktan sonra kendi HTTP uygulamanizi yasmak icin gerekli olan nerdeyse her seyi biliceksiniz.Ozellikle su kisimlar anlasilacak.

1-HTTP mesajlarinin akisi

2-HTTP mesajlaranin uc bolumu(baslangic satiri,headerlar,ve body)

3-Istek ve cevap mesajlari arasindaki farklar

4-Istek mesajlarini destekleyen cesitli fonksiyonlar(mesajlar)

5-Cevap mesajlariyla donen cesitli status kodlari

6-HTTP headerlari ne yapar

MESAJLARIN AKISI

HTTP mesajlari HTTP uygulamalari arasindaki blok seklinde yollanan datalardir.Bu blok datalari meta-information dedigimiz mesajin icerigini ve anlamini tarif eden bir text ile baslar.Sonra opsiyonel data gelir.Bu mesajlar istemci,serverlar ve proxyler arasinda akar.Terimler gelen(inbound),giden(outbound),akinti tersine(upstream),akintiyla beraber(downstream),mesaj yonlerini tarif eder.

GELEN MESAJLARI ANA SERVERA TASIMAK

HTPP inbound ve outbound terimlerini kullanir islem yonlerini tarif edebilmek icin.Mesajlar ana server seyahat ederek gelir(inbound),ve isleri bitti zaman kullaniciya(browsera)geri doner(outbound).

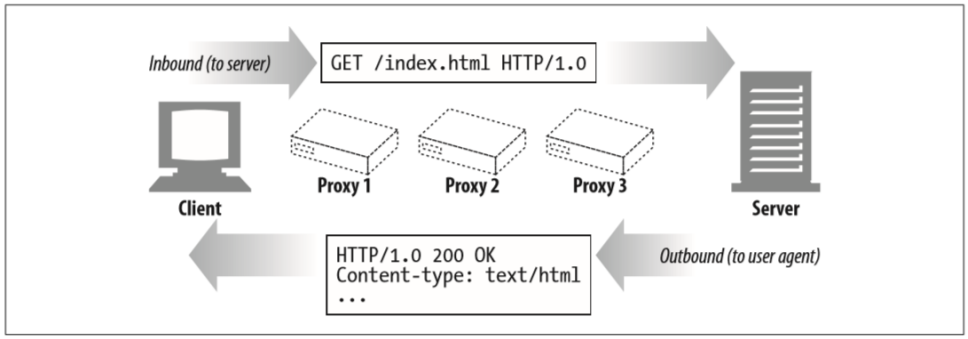

Figure 3-1. Messages travel inbound to the origin server and outbound back to the client

AKINTIYLA AKAN MESAJLAR

HTTP mesajlari nehir gibi akar.Her mesaj akintiyla akar,ne olursa olsun bunlar istek mesaji yada cevap mesajidir(Asagidaki resimde goruceksiniz).Yollanan mesajlar aliciya akinti tersinde gelir gene asagida goruceksiniz.Proxy 1 istek mesajinda proxy 3'e gore akinti tersinde ama proxy 3 te akintiyla ayni yonde.

Mesajlarin Bolumleri

HTTp mesajlari basit,bicimlendirilmis data bloklaridir.Asagidaki resimdeki ornege goz atin.Her mesaj istemciden istek ve serverdan istek mesaji barindirir.Uc bolumden olusurlar:

Baslangic satiri mesaji tarif eder,header bloklari nitelikleiri barindiri,ve opsiyonel body datayi barindirir.

Baslangic satiri ve headerlar ASCII textidir,satirlar ile ayrilir.Her satir iki karakterlik satir sonu sekansi ile biter,satir basi karakteri(ASCII 13) ve satir besleme karakteri (ASCII 10).Bu satir sonu sekansi "CRLF" seklinde yazilir.CR carriage return'dan(satir basi) LF line-feed'den(satir besleme) gelir.HTTP tarifinde satirlari sonlandirmak icin CRLF kullanilir,ve bu degerleri bir isarettir.Bazi eski veya bozulmus HTTP uygulamalari satir basi ve satir besleme olayini kullanmaz.Guclu bir HTTP uygulamasi icin bunlar Kabul edilmelidir.

Body basitce data yiginindan olusur.Baslangic satiri ve headerlarin aksine,body text veya binary data icerebilir.Hicbir sey de icermeyebilir.

Asagidaki ornekde goruceginiz gibi,headerlar body hakkinda birazcik bir bilgi sunar bize.Content-Type satiri body'nin ne oldugunu ornek olarak plain-text.Content-Length satiri bize bodynin ne kadar yer kapladigini soyler.

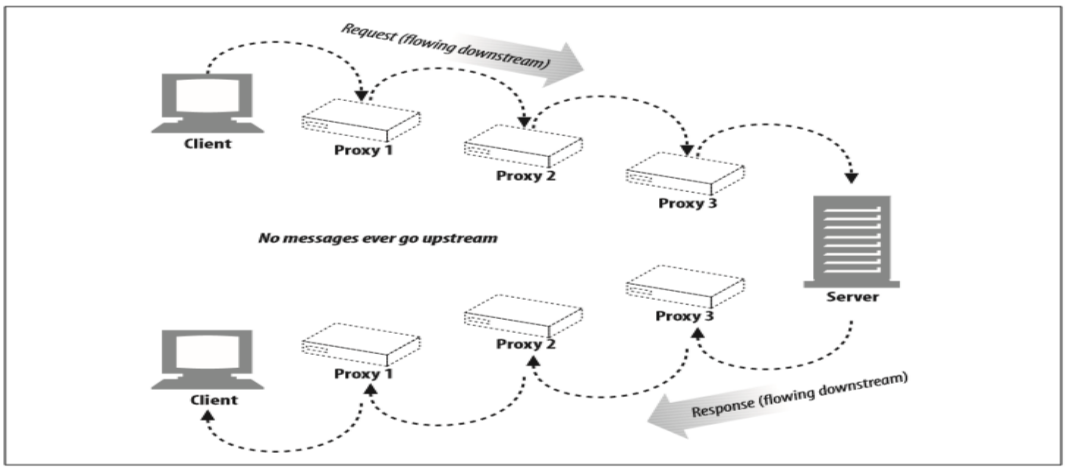

Figure 3-2. All messages flow downstream

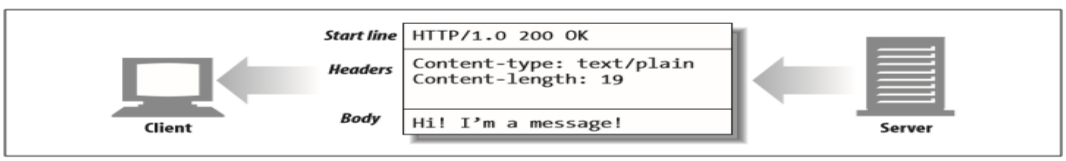

Figure 3-3. Three parts of an HTTP message

Mesaj Syntaxi

Tum HTTP mesajlari iki sekilde bulunur:istek mesajlari ve cevap mesajlari.Istek mesajlari istek olayinda bulunur servera.Cevap mesajlari da istegin sonuclarini istemciye tasir.Iki mesaj tipide ayni basit mesaj yapisina sahiptir.Asagidaki resim istek ve cevap mesajlarini gosteriyor bir GIF imaji icin.

Istek mesajinin formati:

‹method› ‹request-URL› ‹version›

‹headers›

‹entity-body›

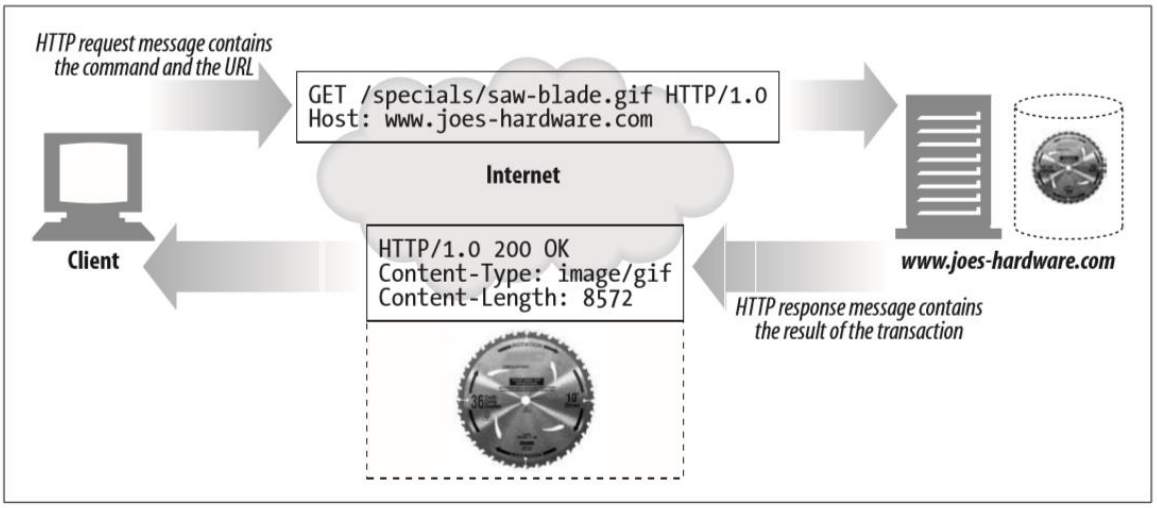

Figure 3-4. An HTTP transaction has request and response messages

Cevap mesaji formati:

‹version›‹status›‹reason-phrase›

‹headerler›

‹entity-body›

Bolumlerin hizlica aciklanmasi:

Method

Istemci server uzerinde kaynak ile oynamak ister.Bu aksiyon tek kelimeyle ifade edilir."GET","HEAD","POST" gibi.Methodu daha sonraki bolumlerde detayli incelicez.

Request-URL

Tum URL istenen kaynagi isimlendirir,veya sadece yol bileseni.Eger server ile direct olarak konusuyorsan,URL'nin yol bileseni genellikle kaynak tam olarak gosterildigi surece okeydir.Server kendisini URL'nin host/portu gibi ustlenebilir.Bolum 2 URL leri kapsar.

Version

HTTP versiyonun mesaj kullanir.format su sekilde gorunur.

HTTP/‹major›.‹minor›

Major ve minor ikiside tamsayidir.HTTP versiyonlamada bunun hakkinda konusucaz.

Status-Kodlari

Uc tane rakamla tarif edilir,istek sirasinda neler oldugu aciklanir.Ilk rakam genel durumdan bahseder("basarili","hata",vb.)Genis bir listeyle status kodlarinin durumu ve anlami ilerki bolumlerde tanimlanacaktir.

Reason-phrase

Status kodunun okunabilir versiyonudur.Satir sonu sekansina kadar tum texti kapsar.Ornekler ilerki bolumlerde verilecektir su an kisa kisa bilgiler veriliyor.Yalnizca insan tarifidir,ornek,"HTTP/1.0 200 NOT OK" ve "HTTP,1.0 200 OK" ikiside basari durumu gibi davranmalidir,fakat buna ragmen reason-phrase diger turlusunu onerir.

Headerler

Sifir veya daha fazlar header,her biri isme sahip,sonra ":" isareti geliyor,sonra opsiyonel olarak bosluk,sonra deger,sonra CRLF.Headerler (CRLF) ile sonlanir,headerin sonun isaretler ve body kismi baslar.HTTP'nin bazi versiyonlarinda,HTTP/1.1 gibi,Istek ve Cevap mesajlarinin olabilmesi icin kesin gerekli headerlarin sunulmasi gerekliydi.Cesitli Headlerdan sonradan bahsedicez.

Body

Body block seklindeki keyfi datayi barindirir.Data olmak zorunda degildir yani.Tum mesajlar body'ye sahip degildir,yani bazen mesaj sadece CRLF ile sonlarinir.Bolum 15'te hepsini tartisicaz.

Simdi istek ve cevap mesajlarinin varsayim seklindeki bir resmini gostericez.

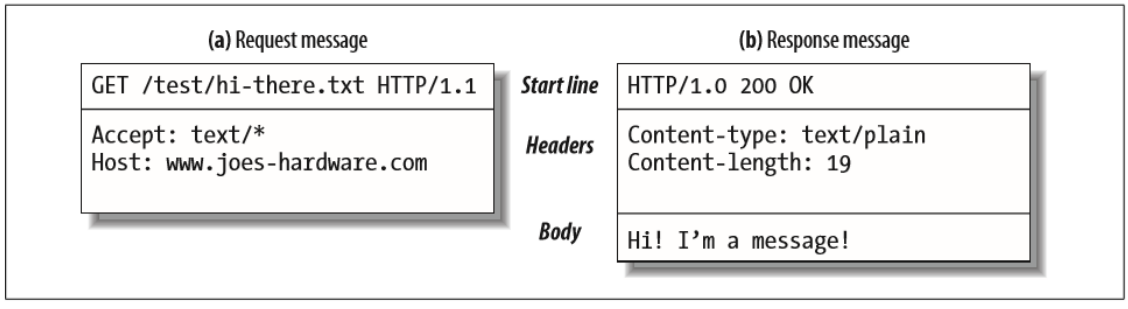

Figure 3-5. Example request and response messages

Sunu bilinki HTTP headerlari herzaman bos satir yani (CRLF) ile biter,hicbir header olmasa bile hicbir body olmasa bile,Gecmise bakildiginda,cogu istemci ve server body yoksa CRLF yi ihmal ediyor bu yanlis.Bazi populer birlikte calisan uyumsuz uygulamalar istemci ve serverlarda CRLF'siz mesajlari Kabul ediyordu.Hackerone'da cok oldugu gibi CRLF

onemli yoksa bir web guvenligi sikintisi cikar ve saldirgan mesajla oynayarak malicious(zararli) hale getirebilir.

Baslangic Satirlari

Tum HTTP mesajlari baslangic satiri ile baslar.Istek mesajindaki baslangic satiri ne yapilacagini,cevap mesajindaki baslangic satiri ne oldugunu soyler.

ISTEK SATIRI

Istek mesajlari kaynakla bir seyler yapmak icin server sorar.Istek mesajlarindaki baslangic satiri veya istek satiri,method icerir;method serverin hangi operasyonu gerceklestirecegini tanimlar ve istek satiri,istek URL si icerir,istek URL'si methodun kaynakta nereyle hasir nesir olacagini soyler.Istek satiri ayni zamanda HTTP versiyonunu kapsarki hangi versiyonla konusulacagi bilinsin.

Bunlarin hepsi beyazbosluk denilen(whitespace) ile ayrilir.Yukardaki Resimde goruceksiniz,istek methodu GET,istek url'si /test/hi-there.txt,ve versiyon HTTP/1.1.

HTTP/1.0'dan once,istek satirlari HTTP versiyonu icermiyordu.

CEVAP SATIRI

Cevap mesajlari status bilgisi ve istemciye sonuclanan veriyi tasir.Cevap mesaji icin baslangic satiri,yani cevap satiri,HTTP versiyonu,status kodu ve text seklindeki yukardada dedigimi reason-phrasi kapsar.

Tum bu alanlar whitespace ile ayrilir gene.Yukardaki resimde goruceksiniz,HTTP versiyonu HTTP/1.0,status kodu 200(success),reason-phrase OK.Yani documenta basarili sekilde ulasildi.HTTP/1.0 dan once cevaplar baslangic satiri icermiyordu.

METHODLAR

Methodlar isteklerin baslangic satirinda baslar,servere ne yapacagini soyler.Ornek,satirda "GET/specials/saw-blade.gif HTTP/1.0" method GET'tir.

HTTP ile ugrasanlar istek methodlari tanimlamislardir.Ornegin ,GET methodu serverdan documenti getirir,POST methodu datayi servera islenmesi icin yollar,ve OPTIONS methodu serverin yapabileceklerini belirler.

Tabloda goruceksiniz 7 tane method var.Sunu bilin ki bazi methodlar istek mesajinda body kismina sahip,ve digerleri bodysiz bebegim evet bebegim evet bodysiz.

Method	Description	Message body?
GET	Get a document from the server.	No
HEAD	Get just the headers for a document from the server.	No
POST	Send data to the server for processing.	Yes
PUT	Store the body of the request on the server.	Yes
TRACE	Trace the message through proxy servers to the server.	No
OPTIONS	Determine what methods can operate on a server.	No
DELETE	Remove a document from the server.	No

Tum serverlar yedi methoduda uygulamaz.Ayrica,HTTP kolayca genisletebilir sekilde tanimlanmistir,diger serverlar kendi istek methodlarini ayriyetten uygulayabilir.Bu eklenen methodlar **genisletilmis method** diye adlandirilir,cunku HTTP'de tanimlananlara gore genisletilmistir.

STATUS KODLARI

Methodlarini server ne yapacagini dedigi gibi,status kodlarida istemciye ne olup bittigini soyler.Status kodlari cevaplarin baslangic satirinda belirirler.Ornegin,satirda "HTTP/1.0 200 OK" status kodu 200.

Istemci istek mesajini HTTP serverina yolladigi zaman,cogu sey degisir.Eger sansliysaniz,istek tamamiyla basarili olacaktir.Her zaman sansli olucaz diye bir sey yok.Server istediginiz kaynagin bulunamadigini soyleyebilir,kaynaga erisiminiz icin yeterli yetki yoktur,yada kaynak baska bir yere tasinmis olabilir.

Status kodlari cevap mesajlarinda ilk satir ile gelir.Rakamlarla ve okunabilir sekilde gelir.Rakamlarla gelen kod programlar icin hata kodlari kolaydir,insanlar icinse okunabilir kisim.

Cesitli status kodlari rakamlarina gore siniflandirilmistir.Status kodlari 200 ile 299 arasi basarili.300 ve 399 arasi tasinmis.400 ve 499 arasi istemci istek mesajinda bir sikinti var.500 ve 599 serverda bir sikinti var.

Resme bakin simdi.

Table 3-2. Status code classes

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

HTTP'nin varolan versiyonlari sadece birkac status kategorisini tanimlar.Protokol gelistikce,daha fazla status kodu HTTP icin calisanlar tarafindan tanimlanicak.Eger status kodu alip ve tanimassaniz,ihtimaller icin su an biri genisletilmis protocol kullaniyor.Genel kullanici olarak neyin neye denk geldigini bilmelisiniz artik.

Ornegin,aldiginiz status kodu 515 ise,bunu bir server hatasi oldugunu bilmelisiniz.5XX ailesine ait sonucta.

Asagidaki tablo en cok karsilastiginiz status kodlarini gostericek.

Table 3-3. Common status codes

Status code	Reason phrase	Meaning
200	OK	Success! Any requested data is in the response body.
401	Unauthorized	You need to enter a username and password.
404	Not Found	The server cannot find a resource for the requested URL.

REASON-PHRASELER

Baslangic satirinin son bilesenidir,status kodlarinin text halini saglar.Ornegin,satirda"HTTP/1.0 200 OK" reason phrase OK'dur.

Reason phraseler status kodlari ile uyumludur.Okunabilirlik saglarki siradan kullanicilar bile istek sirasinda neler olup bittigini anlasin.

HTTP uzerinde calisanlar kesin olarak soyle boyle olucak diye bir kural kanun falan tanimlamamislar.Sonraki bolumlerde,status kodlari ve onerilen reason-phraseleri koyucaz.

VERSIYON NUMARALARI

Versiyon numaralari istek ve cevap mesajlarinin baslangic satirlarinda gorulur.HTTP/x.y sekli.HTTP uygulamalarinin birbirlerine hangi versiyon protokolunun uygun oldugunu soylemesini saglar.

Versiyon numaralari uygulamalarin HTTP ile konusurken birbirlerinin kapasitelerini ve mesaj formatlarini bilmesi amaciyla tasarlanmistir.HTTP versiyon 1.2 uygulamasi HTTP 1.1 uygulamasi ile haberlesirken 1.2'deki yenilikleri kullanmadigini bilmesi gerek,uygulamalar bu yuzden olabildigince yasli versyionlari ile konusmazlar.

Versiyon numarasi uygulamanin destekledigi en yuksek HTTP versiyonunu kapsar.Bazi durumlarda bu uygulamalar arasinda karisikliga yol acar,cunku HTTP/1.0 uygulamalari cevabi yorumlarken HTTP/1.1 sekliyle ki cevap 1.1 cevabi olsun.Bu sadece uygulamalari cevaplerken protocol level farkliligi durumunda kullanilir.

Sunu bilinki versiyon numaralari kesirli sayi falan degildir.Versiyondaki her Numara islenir.Yani HTTP versiyonlarini karsilastirirken,her Numara dikkatlice karsilastirilmalidirki hangisi daha yuksek versiyon karar verilsin.Ornegin,HTTP/2.22 HTTP/2.3'den yuksektir,cunku 22>3 ten.

HEADERLAR

Onceki bolumde istek ve cevap mesajlarindaki ilk satira odaklandik(methodlar,status kodlari,reason-phraseler,versiyon numaralari).Sonra 0 veya 1 veya bircok HTTP headeri gelebilir.Yukarlara cikin resimde cevap mesajinda headerlari goruceksiniz.

HTTP header Alani ayriyeten bir bilgi saglar,hem istek hemde cevap mesajlari icin.Basitce isim/deger olayi vardir.Ayni URL'lerdeki sorgu veya parametre olayindaki gibi.Ornek Content-length headeri;

Content-length: 19

HEADER SINIFLANDIRMALARI

HTTP uzerinde calisanlar tarafindan birkac header Alani tanimlanmistir.Uygulamalar ayni zamanda kendi ev yapimi headerlarini icat edebilirler.HTTP headerlari soyle siniflandirilir.

GENERAL HEADERLER

Istek ve cevap mesajlarinda gorunebilir.

ISTEK HEADERLARI

Istek hakkinda daha fazla bilgi saglar

CEVAP HEADERLARI

Cevap hakkinda daha fazla bilgi saglar

VARLIKLA ALAKALI HEADERLAR(ENTITY HEADER)

Body kisminin boyutunu,icerigini veya kaynagin kendisini tarif eder.

GENISLETILMIS HEADERLER

Yeni headerlardir,HTTP tanimindan tanimlanmamislardir.

Her HTTP headeri basit bir syntaxe sahiptir:isim,":",opsiyonel bosluk,deger,CRLF.Asagidaki tabloda ornek headerlari goruceksiniz simdi.

Table 3-4. Common header examples

Header example	Description
Date: Tue, 3 Oct 1997 02:16:03 GMT	The date the server generated the response
Content-length: 15040	The entity body contains 15,040 bytes of data
Content-type: image/gif	The entity body is a GIF image
Accept: image/gif, image/jpeg, text/html	The client accepts GIF and JPEG images and HTML

HEADER DEVAM SATIRI

Uzun header satirlarini daha okunabilir yapmak icin bircok satira ayrilir.Extra satir yapilarak bir bosluk veya tab karakteri bosluk birakilir.

Ornegin:

HTTP/1.0 200 OK

Content-Type:image/gif

Content-Length:8572

Server: Test Server

Version 1.0

Bu ornekte,cevap mesaji Server headeri devam satirina sahip.Tum deger aslinda "Test Server Version 1.0".

Tum headerleri sonraki bolumde incelicez.Appendix C bolumunde detayli header referans ornegi vericez.

BODY KISMI

3.kisim olarak HTTP mesajlari opsiyonel body kismina sahiptir.Body kismi HTTP mesajlarinin yuk kismidir aslinda.HTTP icin en basinda kurye seklini ornek vermistik ya Kargo arabasi dusunun icindeki yukler iste.HTTP'de tasinir bu body kismi.

HTTP mesajlari bircok cesit digital data tasiyabilir:images,video,HTML documents,yazilim uygulumalari,kredi karti islemleri,elektronik mail,ve devam eder bole.

VERSIYON 0.9 MESAJLARI

HTTP versiyon 0.9 HTTP protokolunun ilk versiyonlarindan.O zamanlarda istek ve cevaplariyla basladi.Bugunku HTTP de olan basit bir ornek hemen.

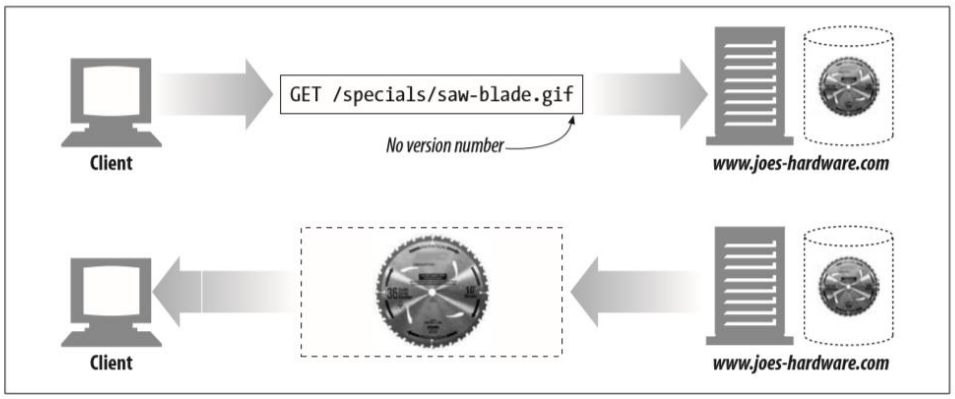

Figure 3-6. HTTP/0.9 transaction

HTTP/0.9 mesajlari istek ve cevaplarida barindirir.Ama istek sadece method ve request URL'yi barindiyor.Cevap ise sadece body'yi barindiriyor.Ne versiyon bilgisi,ne status kodu,ne reason-phrasi ne de header hicbiri yok.Harbi iyi ki gelistirilmis.HTTP/0.9'u gelistirenler tek burda kalmamis gelistirmeye devam etmisler.Para icin mi.Hayir.Itibar icin mi.Hayir,neden peki.Su anda disarda gezmek tozmak yerine oturup bir insan niye calisir.Su anda yapilanlar sayesinde her seyin tabaninda web var yani HTTP her yerde su anda.Cunku onlar icin onemli olan insanliga bir seyler katmak gecmisteki insanlarin yaptiklarini omuzlarina alarak onlari devam ettirmek.Insanlara karsi umidimiz olmayabilir ama hala insanogluna karsi var.Neyse devam edelim yoksa baya yazarim burda.

Ancak,Bu basitlik

Esneklik veya gorevi yerine getirmek konusunda bu kitapta tarif edilen cogu HTTP yeniligine ve uygulamasina izin vermemistir.Kisaca burda tarif ettik cunku hala istemciler,serverlar ve diger uygulamalar kullaniyor.Uygulamayi yazanlar 0.9 versiyonun limitlerinin farkinda olmalilar.

METHODLAR

HTTP methodlarinin basicleri hakkinda biraz daha konusalim.onceden listelenmisti yukarda.Sunu unutmayin her method her server tarafindan uygulanmaz.Itaatkar HTTP Version 1.1'imizde,server sadece GET ve HEAD methodlarini uygulamali.

Hatta serverlar tum methodlari uygulamaya koysa bile,bu methodlar kisitli sayida kullanir.Ornegin,server DELETE veya PUT(sonradan konuscaz bu methodlarini)kullaniyor.Herhangi birinin dosyalarinizi silmesini veya saklamasini istemessiniz degil mi.Bu kisitlamalar genelde server ayarlari yapilirken kurulurki siteden siteye ve serverdan server degisirmi kontrol edilsin.

GUVENLI METHODLAR

HTTP bazi methodlari guvenilir method olarak tanimladi.GET ve HEAD methodlari guvenilir deniyor.Bu demektirki GET veya HEAD methodlari ile islem yapan HTTP istekleri sirasinda hic bir sikinti meydana gelmeyecek.

Hicbir sikinti meydana gelmeyecek derken,sunu kast ettik HTTP istegiyle beraber serverda hicbir sey olmayacak.Ornegin,dusununki Joe's Hardware sitesinden alisveris yapiyorsunuz.Satin al buttonuna tikliyorsunuz.Butona tiklamak POST methoduyla istek yapmak demekki bu kredi karti bilgileriniz demek(Bunu sonraki kisimda tartisicaz).Ve olay sizin adiniza serverda gerceklesiyor.Bu durumda,Olayda kredi kartiniz satin alma olayi icin tehsil ediliyor.

Guvenli methodlar kullanildigi zaman hic bir sikinti olusmayacak diye bir garanti yok.(Pratikte bu web developerlarinin isi).Guvenli methodlar su anlama geliyor.HTTP uygulama developerlari kullanicilarin guvesniz method kullanarak islemler yaptiklarini bilmesini saglamak.Joe Hardware ornegindeki gibi,web tarayiciniz bir anda uyari mesajiyla belirebilir bu sizin bir istekte bulundugunuzu ve guvensiz method kullandiginizi gosterir.Bunu sizde anlarsiniz.Sonuc olarak serverda bir seyler olabilir(kredi kartinizin calinmasi gibi vb.).

GET

En cok kullanilan methoddur.Kaynagi yollamak icin servera sorar.HTTP/1.1 serverlari bu methodu kullanmasi gereklidir.Resimde goruldugu gibi istemci GET methoduyla HTTP istegi yapiyor.

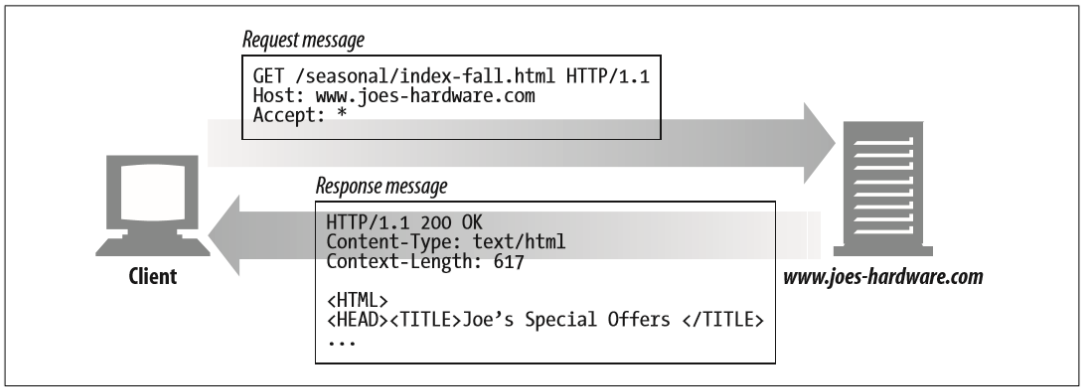

Figure 3-7. GET example

HEAD

Head methodu nerdeyse GET ile aynidir,ama sadece header kismi doner,body kismi donmez.Buda kaynagi getirmeden headerleri denetlememize izin verir.HEAD'I kullanarak sunlari yapabilirsiniz:

1-Kaynagi getirmeden kaynak hakkinda bilgi getirebilirsiniz.

2-Eger bir obje varsa ortada,cevabin status koduna bakabilirsiniz.

3-Headerlara bakarak kaynakla oynanmis mi test edebilirsiniz.

Server developerlari sunu saglamaliki headerlar GET methodunda nasil geliyorsa ayni sekilde HEAD'de gelmeli.HEAD methodu ayni zamanda HTTP/1.1 'e uymak icin gereklidir.

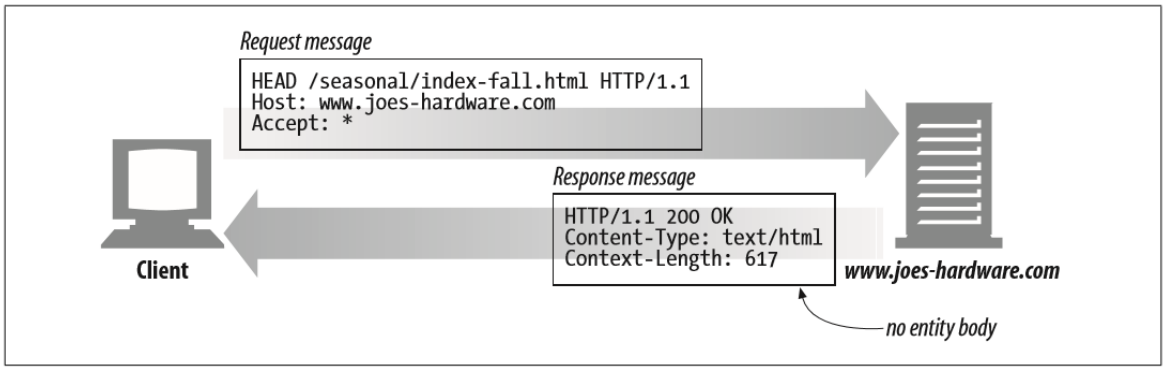

Figure 3-8. HEAD example

PUT

PUT method documentleri servera yazar,GET methoduyla documentleri serverdan okuyorduk onun tersidir.Bazi acik sistemler site sayfasi yaratmaniza ve onu web serverina yuklemenize izin verirler.

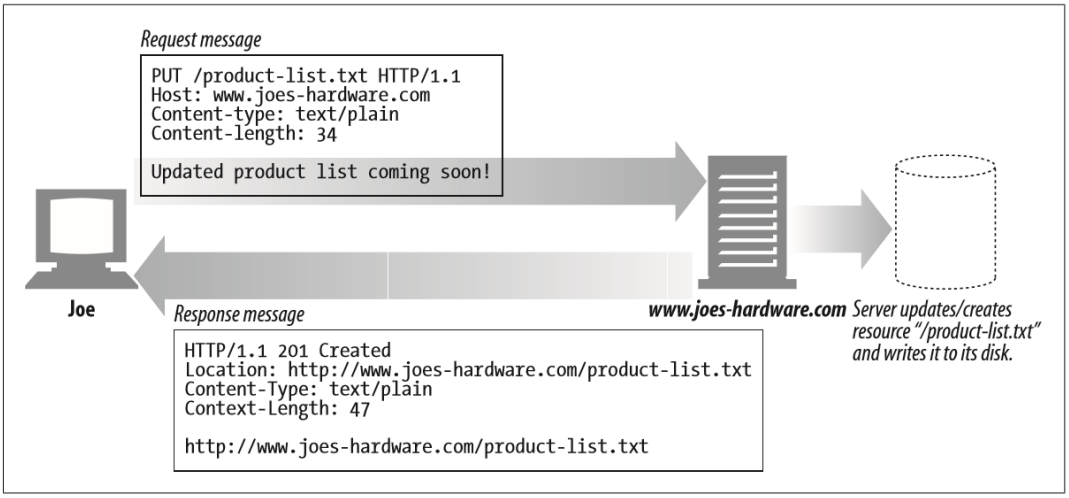

Figure 3-9. PUT example

PUT methodunun olayi su:Yapilan istekte server istegin body kismini alir,birde request-URL kismini alir ve o URL adresinde istegin bodysi seklinde bir document olusturulur,URL'de bir dosya varsa,body kismi degistirilir.

Cunku PUT icerigi degistirmenize izin veren bir yapidir,bu yuzden cogu web server PUT methodunu kullanmaniz icin password ile login olmanizi gerektirir.

POST

Post methodu input edilmis datayi server yollamak icin kullanilir.Pratikte,siklikla HTML formlarinda kullanilir.Doldurulmus form'a sahip olan data genelde servera yollanir,sonra nereye gitmesi gerekiyorsa oraya yonlendirilir(misal server gatewayine).Asagidaki resim istemcinin HTTP istegi ile form datasini servera POST methoduyla nasil yolladigini gosteriyor.

TRACE

Istemci istek yaptigi zaman,istek guvenlik duvarlari,proxyler,gatewayler,veya diger uygulamalar aralarindan dolasabilir.Bunlarin hepsi HTTP istegini degistirebilir.TRACE methodu su ise yarar,Istemci istegi yaptigi zaman istek server ulastigi zaman sonunda nasil gozukur olay bu.

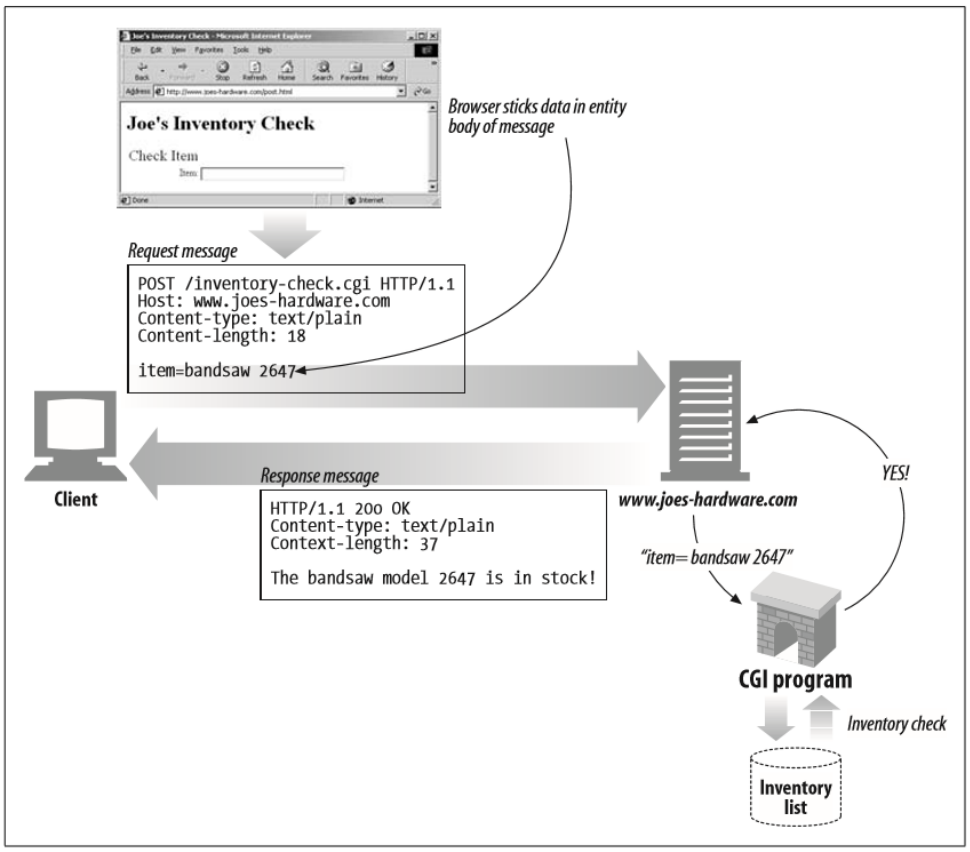

Figure 3-10. POST example

TRACE istegi hedef serverda "loopback" arizasi baslatir.Serverdaki son bacaktan hooooop TRACE cevabi geri doner,el degmemis istek mesajimiz cevabin body kismina alinir.Istemci bundan sonra orjinal mesaj istek/cevap zincirlerinin herhangi bir yerinde degistirildi yada kurcalandiysa bunu anlayacaktir.

TRACE methodunun ana amaci hata kontroludur.Istek/cevap olaylarini dogrularsiniz bu methodla,ayrica proxylerin ve diger uygulamarin etkilerini gormek icin iyi bir aractir.

TRACE hata kontrolu icin iyi oldugu kadar,Uygulamalara mudahale ederek baska methodlar gibi davranmasi sakincalidir.Cogu HTTP uygulamasi methoda bagli olarak farkli seyler yaparlar,Ornegin,proxy POST istegini direkt olarak gecirecektir ve server ulastiracaktir,ama GET methodu yollamaya kalkarsaniz baska bir HTTP uygulamasina yonlendirecektir direk gecirmeyecektir mesela web onbellegine yollamak gibi.TRACE bize methodlari birbirinden ayirmamizi saglamamaz,genelde aradaki uygulamalar TRACE isteginde ne gibi kararlar aliyor ona bakilir.

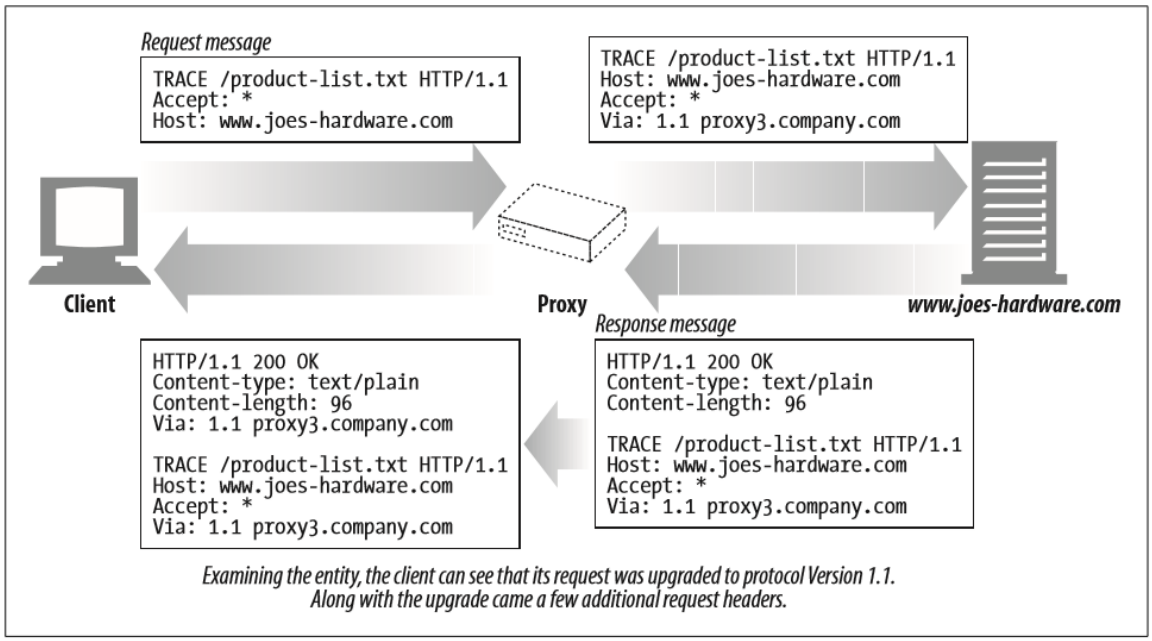

Figure 3-11. TRACE example

BODY kismi TRACE isteginde yollanmaz,BODY kismi TRACE cevabinda kelimesi kelimesine istegi alan serverin cevabinda icerir.Yani serverin cevabinda icerir.Pek anlatamadim ama yukardaki resme bakin istek mesajindaki her sey cevap kismindaki headerlardan sonra bulunuyor kelimesi kelimesine duruyor orda.Arada da headerlarla body'yi ayiran CRLF var.

OPTIONS

OPTIONS methodu server sen neler yapabiliosun kardesim goster bana bakayim der.Destekledigi methodlari falan her seyi sorar.

Bu Istemci Uygulamasi icin iyi bir sey cunku kaynaga en iyi sekilde nasil ulasirim onu gosteriyor.

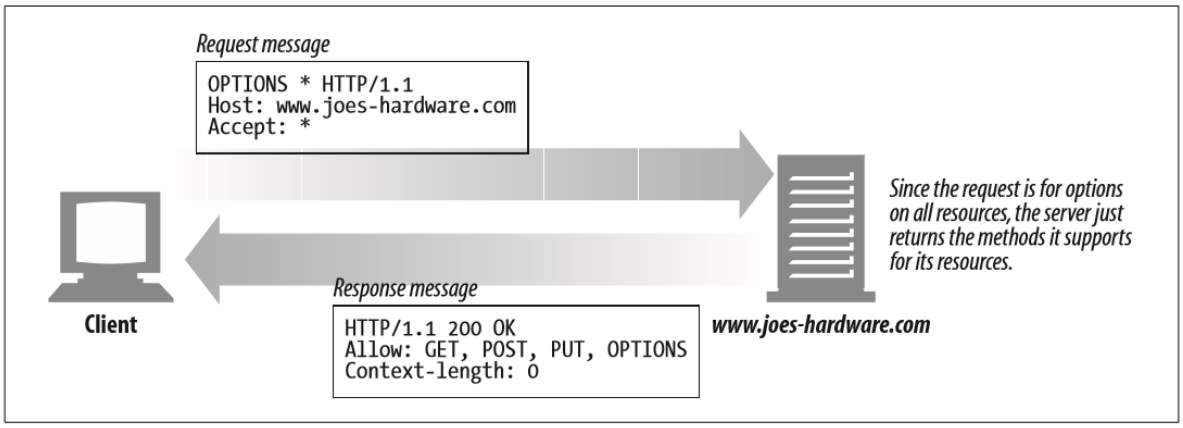

Figure 3-12. OPTIONS example

DELETE

DELETE methodu dusundugunuzu yapiyor.Aynen tutup yapiyor.Servera kaynaktaki dosyayi silmek icin soruyor.Ancak,istemci uygulamasi sanmasinki kesin olucak diye bir sey yok.Cunku HTTP'yi tasarlayanlar serverin istegi istemciye soylemeden gecersiz kilma hakki tanimislar.DELETE methodunun ornegi asagida olucak.

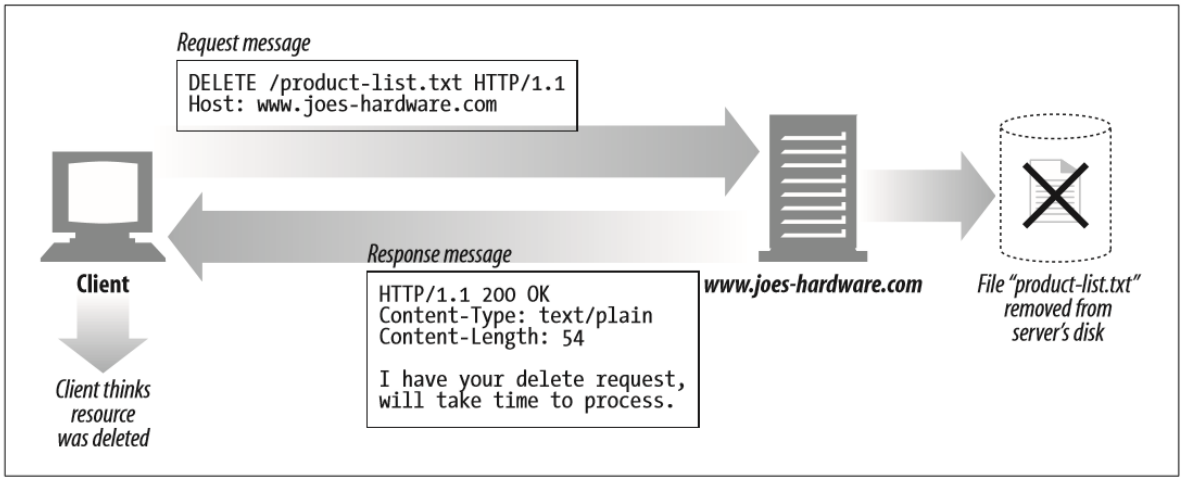

Figure 3-13. DELETE example

GENISLETILMIS METHODLAR

HTTP genisletilebilir bir alan olarak tasarlandi,yani yenilikler eski yazilimin hataya sebep olmasina neden olucak diye bir sey yok.Genisletilmis methodlar HTTP/1.1 da tasarlanan methodlar degil.Developerlara HTTP servislerini genisletebilme ve onlari uygulayarak serverlari yonetme olanagi saglar.Bazi kullanilan ornek genisletilmis methodlar asagidaki tabloda var.Bu methodlar WebDAV HTTP genisletilmis methodlaridir.Web iceriklerinin web serverlarina HTTP uzerinden yayilmasina yardimci olur.

Table 3-5. Example web publishing extension methods

Method	Description
LOCK	Allows a user to "lock" a resource—for example, you could lock a resource while you are editing it to prevent others from editing it at the same time
MKCOL	Allows a user to create a resource
COPY	Facilitates copying resources on a server
MOVE	Moves a resource on a server

Onemli bir sey diyecegim aklinizda bulunsun,tum genisletilmis methodlar resmi degildir,Yani sizde genisletilmis bir method tasarlayabilirsiniz,ama cogu HTTP uygulamasi anlamayacaktir.Ayni zamanda methodu kullanan HTTP uygulamasi baska uygulama ile haberlesirken diger uygulama anlamayacaktir.

Bu gibi durumlarda,en iyisi genisletilmis methodlara tolerans gostermektir.Proxyler gecis mesajlarini bilinmeyen bir methodla server dogru yollayacaktir tabii ki ama server bunu uctan uca yapabilecek kapasiteye sahip ve kirilmayacaksa gerceklesir.Aksi takdire,501 status kodu ile cevap verilmesi gerek (Uygulanamadi).Genisletilmis versiyonlarla ugrasabilmek icin en iyi yol eski kuraldir.Atalarimiz soyle diyor:"Yolladiklarinda tutucu ol,Kabul ettiklerinde ozgur ol."

STATUS KODLARI

HTTP status kodlari 5 kategoriye ayriliyor.Yukarlarda gostermistik aslinda ama simdi daha cok uzerinde durucaz.

Status kodlari islemlerin sonucun anlamalari icin kolay bir yok.Bu bolumde,ayni zamanda reason-phraseleride siralicaz,ama tabiiki reason-phrase icin kesin bir metin olayi yok.HTTP/1.1'deki reason-phrase leri oneriyoruz.

100-199:BILGI STATUS KODLARI

HTTP/1.1 bilgi status kodlariyla giris yaptik suan,aslinda bu status kodlari nispeten yeni ve karisiklik,deger saglama konularinda azcik tartismaya acik.Status kodlarimiz gelsin simdi.He birde 100-199 dedik diye 99 tane kod gostericez diye bir sey yok he araligin olayini bilin yeter.

Table 3-6. Informational status codes and reason phrases

Status code	Reason phrase	Meaning
100	Continue	Indicates that an initial part of the request was received and the client should continue. After sending this, the server must respond after receiving the request. See the Expect header in Appendix C for more information.
101	Switching Protocols	Indicates that the server is changing protocols, as specified by the client, to one listed in the Upgrade header.

100 Continue status kodu,ozellikle biraz kafa karistirici olabilir.HTTP istemci uygulamasi servera body kismini yollar ama yollamadan once serverin body'yi Kabul edicegini ogrenmek ister,iste bu yuzden 100 kdou vardir.Daha sonra detaylari ile tartisicaz.

Istemci ve 100 CONTINUE

Eger istemci body'yi server yollayip,100 CONTINUE cevabini beklemek istiyorsa,body'yi yollamadan once,istemci istek headerini yollamasi gerek.Bu headerdan bahsedicez sonra.100-contunue.Eger istemci body'yi yollamayacaksa,100-continue headeri yollanmamali,cunku bu sadece serverin kafasinin karisip istemcinin belki body'yi yollamistir sekilde dusunmesine yol acacak.

100-continue,bir cok sekilde,bir optimizasyondur.Istemci uygulamasi 100-continue headerini sadece serverin ilgilenemeyecegi veya kullanamayacagi buyuk boyutlu body'leri yollamasini engellemek icin kullanilmali.

Cunku basta dedigimiz 100 Continue statusu kafa karistirir olayini hakli cikariyor.Istemci 100-continue headerini yolladiktan sonra serverin 100-continue cevabini yollamasini sonsuza kadar bekleyemez.Zaman asimindan sonra,istemci sadece body'yi yollar.

Pratikte,istemci'yi kullanan insanlar beklenmeyen 100 Continue cevaplarini hazirlikli olmali(sikici ama gercek).Bazi les HTTP uygulamalari uygunsuz sekilde bu 100 continue kodunu yollayabilir.

Gencler istemci 100-continue headerini yolluyor derken kafaniz karismasin.Expect headeri denen bir sey var.Sunucudan(serverdan)100-continue cevabini istedigi icin 100-continue headeri koyuyor istegine ve oyle yolluyor server.Bunlara biraz expect header diyoruz yani Beklenen header diyebiliriz.

SERVERLAR VE 100 CONTINUE

Eger server istekte beklenen header:100-continue degerini alirsa,100 Continue cevabini vermeil yada hata kodu yollamali.Asagidaki tabloda gorursunuz.Serverlar asla 100 Continue status kodunu istemciye yollamamali oyle bir beklentide de bekletmemeli istemciyi ama yukarda dedigimiz gibi bazi mal serverlar yapiyor bunu.

Eger herhangi bir nedenden oturu serverlar body kismini 100 Continue cevabini yollamadan once alirsa,status kodu yollamaya gerek yok,cunku istemci zaten devam etmeye karar vermis.Server istegi okumayi bitirdigi zaman,ancak,hala istege karsilik bir status kodu yollamak zorunda(100 Continue status es gecebilir).

Sonunda,eger server istemciden 100-continue header istegini alirsa ve body'yi okumadan once istegi sonlandirmaya karar verirse,sadece cevabi yollayip baglantiyi kapatmamali,istemcinin cevabi almasi onlenmeli(4.Bolumde tartisicaz.)

PROXYLER VE 100 CONTINUE

100-Continue headerini alan proxy'nin bir kac sey yapmasi gerekli.Eger proxy sonraki atlayacagi server biliyorsa,HTTP/1.1 versiyonu yada baska versiyon fark etmez.Versiyonun ne oldugunu bilmiyorsa.100-Continue headerini atlayacagi server aktarir.Eger biliyorsa ve 1.1'den onceyse 417 Beklenti hatasi verilir.

Eger proxy 100-Continue headerini HTTP/1.0 veya daha genc versiyonlarinnin istekte istemci adina eklemeye karar verirse,100 Continue cevabini istemciye iletmemeli.Cunku istemci onunla ne yapacagini bilmez.

Proxylere daha fazla yatirim yapilmali ki serverlara atlarken durumlari surdursun ve HTTP versiyonlarini desteklesin.Boylece istemciden gelen 100-Continue headeriyla daha iyi ilgilenebilir.

Kisaca arkadaslar 100-Continue headeri istemciden geldigi zaman serverlar uzak durmali ve gecistirmeli ayni istanbulda duraklarda herkesten eve gitmek icin 1-2 lira isteyen insanlar gibi uzak durun ve gecistirin eger her turlu geliyorsa hata verin yani para yok aga.

200-299:BASARI STATUS KODLARI

Istemci istek yaptigi zaman,istek genelde basarili olur.Serverlar bircok cesit istekle uyusan bir dizi status basari kodlarina sahiptir.Tablo geliyor hemen.

Table 3-7. Success status codes and reason phrases

Status code	Reason phrase	Meaning
200	OK	Request is okay, entity body contains requested resource.
201	Created	For requests that create server objects (e.g., PUT). The entity body of the response should contain the various URLs for referencing the created resource, with the Location header containing the most specific reference. See Table 3-21 for more on the Location header. The server must have created the object prior to sending this status code.
202	Accepted	The request was accepted, but the server has not yet performed any action with it. There are no guarantees that the server will complete the request; this just means that the request looked valid when accepted. The server should include an entity body with a description indicating the status of the request and possibly an estimate for when it will be completed (or a pointer to where this information can be obtained).
203	Non-Authoritative Information	The information contained in the entity headers (see "Entity Headers" for more information on entity headers) came not from the origin server but from a copy of the resource. This could happen if an intermediary had a copy of a resource but could not or did not validate the meta-information (headers) it sent about the resource. This response code is not required to be used; it is an option for applications that have a response that would be a 200 status if the entity headers had come from the origin server.
204	No Content	The response message contains headers and a status line, but no entity body. Primarily used to update browsers without having them move to a new document (e.g., refreshing a form page).
205	Reset Content	Another code primarily for browsers. Tells the browser to clear any HTML form elements on the current page.
206	Partial Content	A partial or *range* request was successful. Later, we will see that clients can request part or a range of a document by using special headers—this status code indicates that the range request was successful. See "Range Requests" in Chapter 15 for more on the Range header. A 206 response must include a Content-Range, Date, and either ETag or Content-Location header.

300-399: YONLENDIRME STATUS KODLARI

Yonlendirme status kodlari istemciye kaynaga alternatif konumu kullanmasini soyler.Eger kaynak tasindiysa,yonlendirme status kodlari ve opsiyonel konum headeri istemciye yollanirki bulunsun kaynak.Bu istemciye yeni bir konuma yani URL'ye gitmesini saglar hemde isini insanlara birakmadan.

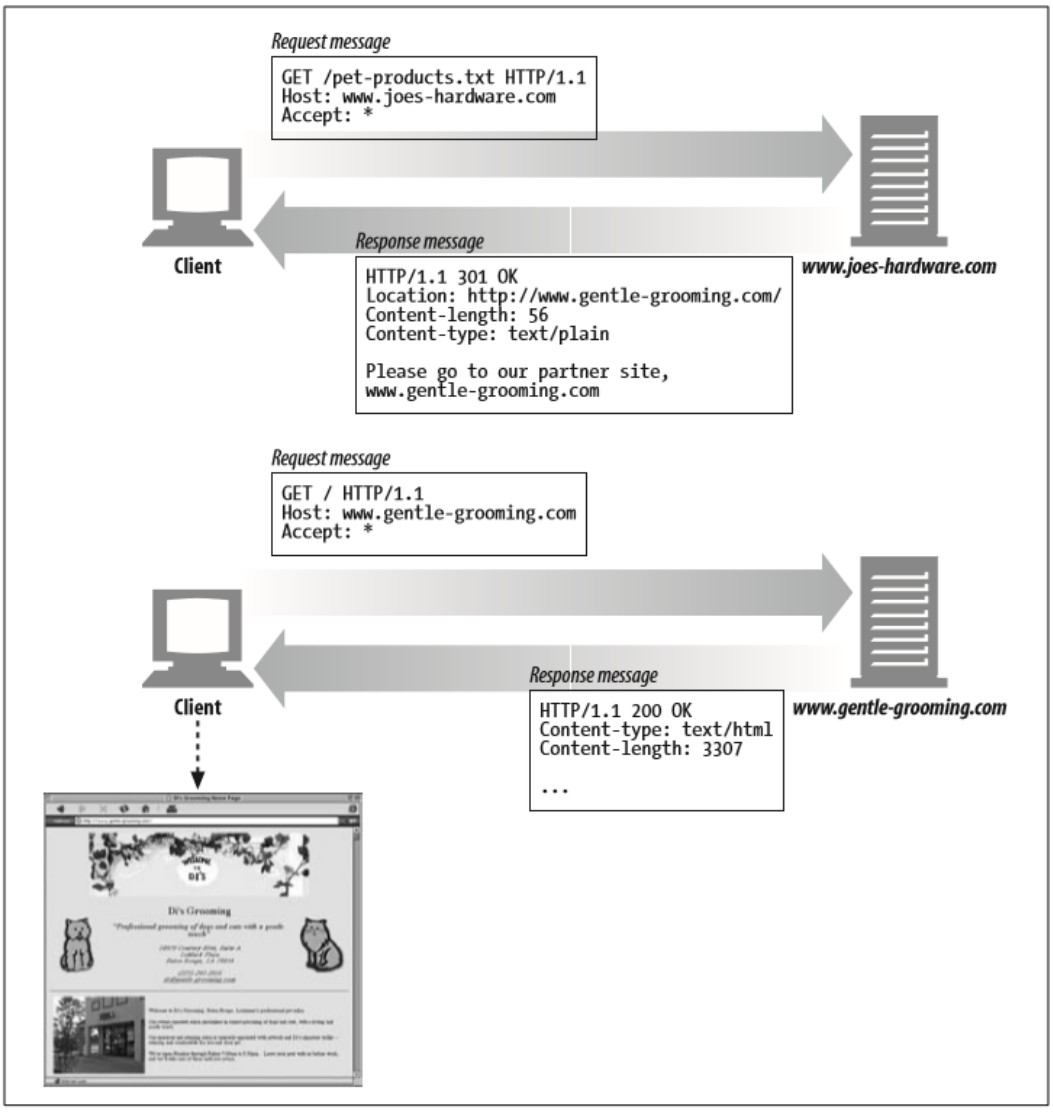

Figure 3-14. Redirected request to new location

Bazi yonlendirme status kodlari kaynagin ana serverindaki kopyasina yonlendirir.Ornegin,bir HTTP uygulamasi kaynagin kopyasi guncel mi yada ana serverdaki kaynak degistirilmis mi diye kontrol edebilir.Asagidaki resimde ornegi var.Istemci If-Modified-Since headerini yolluyor.Bu headerla Ekim 1997'den itibaren degismis dosyalari getir diyor.Eger document o tarihten itibaren degismemisse,server icerik yerine 304 status kodunu yolluyor.

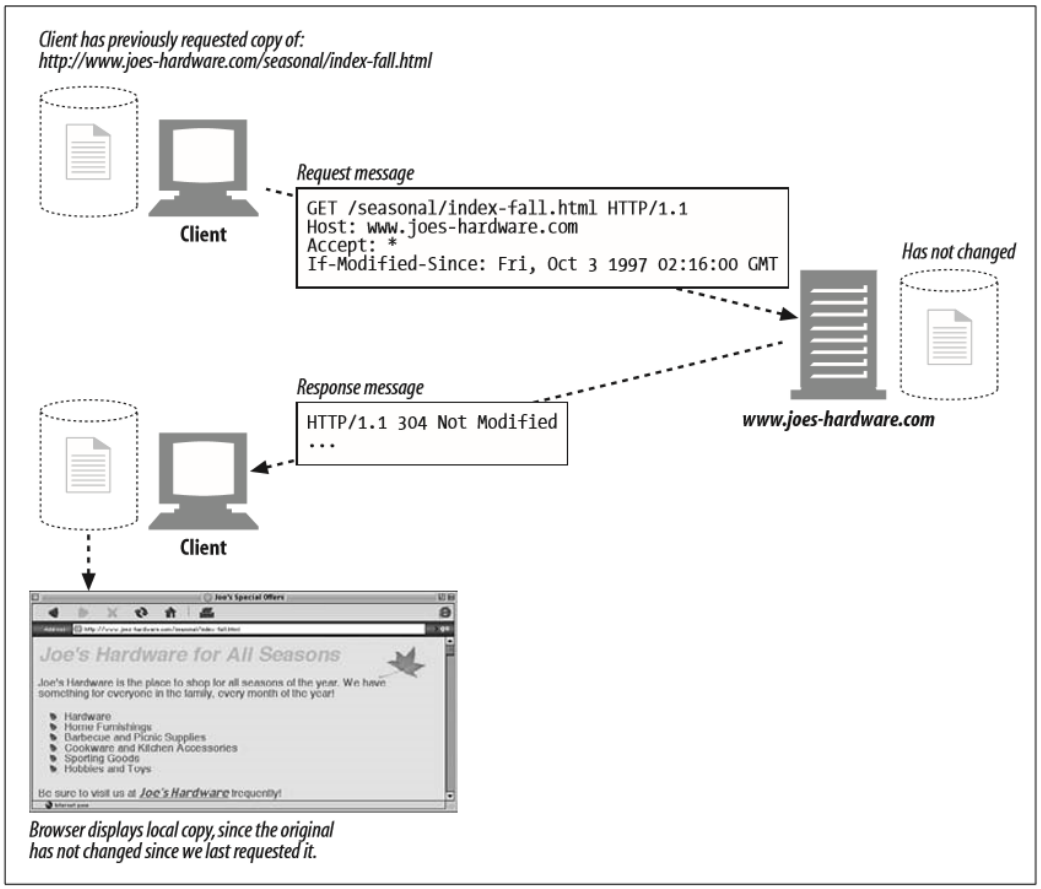

Figure 3-15. Request redirected to use local copy

Genelde,cevaplar icin iyi bir Pratik olmasi acisindan fazladan isteklerde headerlerin olmamsi,yonlendirme status kodlari icin,bodyler icin,yonlendirilmis URL'ler icin iyidir.Figure 3-14. Yazan resimde ilk cevapta gorebilirsiniz.Simdi yonlendirme status kodlarina bakalim.

Table 3-8. Redirection status codes and reason phrases

Status code	Reason phrase	Meaning
300	Multiple Choices	Returned when a client has requested a URL that actually refers to multiple resources, such as a server hosting an English and French version of an HTML document. This code is returned along with a list of options; the user can then select which one he wants. See Chapter 17 for more on clients negotiating when there are multiple versions. The server can include the preferred URL in the Location header.
301	Moved Permanently	Used when the requested URL has been moved. The response should contain in the Location header the URL where the resource now resides.
302	Found	Like the 301 status code; however, the client should use the URL given in the Location header to locate the resource temporarily. Future requests should use the old URL.

Table 3-8. Redirection status codes and reason phrases (continued)

Status code	Reason phrase	Meaning
303	See Other	Used to tell the client that the resource should be fetched using a different URL. This new URL is in the Location header of the response message. Its main purpose is to allow responses to POST requests to direct a client to a resource.
304	Not Modified	Clients can make their requests conditional by the request headers they include. See Table 3-15 for more on conditional headers. If a client makes a conditional request, such as a GET if the resource has not been changed recently, this code is used to indicate that the resource has not changed. Responses with this status code should not contain an entity body.
305	Use Proxy	Used to indicate that the resource must be accessed through a proxy; the location of the proxy is given in the Location header. It's important that clients interpret this response relative to a specific resource and do not assume that this proxy should be used for all requests or even all requests to the server holding the requested resource. This could lead to broken behavior if the proxy mistakenly interfered with a request, and it poses a security hole.
306	(Unused)	Not currently used.
307	Temporary Redirect	Like the 301 status code; however, the client should use the URL given in the Location header to locate the resource temporarily. Future requests should use the old URL.

Yukardaki tablodan,farketmissinizdir 302,303 ve 307 status kodlari biraz ortusuyor.Bunlarin nasil kullanildigi arasinda ince farklar var,kokunde aslinda HTTP/1.0 ve HTTP/1.1 uygulamalarinda bir fark var.

HTTP/1.0 kullanan bir istemci POST istegi yaptigi zaman ve 302 status kodu dondugu zaman,Yonlendirilen URL'ye POST yerine GET istegi yollanacak.

HTTP/1.0 serverlari HTTP/1.0 kullanan istemcilerden sunu yapmalarini bekliyor.HTTP/1.0 serveri 302 status kodunu yolladigini zaman.POST istegi yapan istemci yonlendirildigi URL'ye GET istegi yapsin.

Bu karisiklik HTTP/1.1 ile devam etti.HTTP/1.1 uzerinden calisanlar 303 status kodunu gelistirdi ama ayni davranis sergilendi.303 status kodu yollandiktan sonra yonlendirilen URL'ye GET istegi yapiliyordu.

Bu karisikligin ustesinden gelmek icin,HTTP/1.1 uzerinden calisanlar 307 status kodunu ortaya atti.302 yerine 307 yerini aldi.Istemcileri gecici olarak yonlendiriyordu.Serverlar 302 status kodunu HTTP/1.0 istemcilerine kullanmak icin sakladilar.

Tum bu seyler kaynarken serverin istemciden gelen istegin HTTP versiyonuna bakip onara gore status kodu belirlemesi gerek.

400-499:ISTEMCI HATA KODU

Bazen istemciler serverin ustunden gelemeyecegi isteklede bulunabilir,mesela kotu formadik istek mesaji veya siklikla olmayan bir URL'ye gidilmesi.

Bunlarin hepsini surf yaparken karsimiza cikan cok populer olan 404 NOT FOUND koduyla biliyoruz.Server bize burda istedigimiz kaynak hakkinda hicbir sey bilmedigini soyluyor.

Cogu istemci hatasi tarayicinizla isi vardir,size rahatsiz etmeden,cok azi,404 gibi,bazen hala gecebilir.Cesitli hata kodlariyla tablomuz geliyor.

Table 3-9. Client error status codes and reason phrases

Status code	Reason phrase	Meaning
400	Bad Request	Used to tell the client that it has sent a malformed request.
401	Unauthorized	Returned along with appropriate headers that ask the client to authenticate itself before it can gain access to the resource. See Chapter 12 for more on authentication.
402	Payment Required	Currently this status code is not used, but it has been set aside for future use.
403	Forbidden	Used to indicate that the request was refused by the server. If the server wants to indicate why the request was denied, it can include an entity body describing the reason. However, this code usually is used when the server does not want to reveal the reason for the refusal.
404	Not Found	Used to indicate that the server cannot find the requested URL. Often, an entity is included for the client application to display to the user.
405	Method Not Allowed	Used when a request is made with a method that is not supported for the requested URL. The Allow header should be included in the response to tell the client what methods are allowed on the requested resource. See "Entity Headers" for more on the Allow header.
406	Not Acceptable	Clients can specify parameters about what types of entities they are willing to accept. This code is used when the server has no resource matching the URL that is acceptable for the client. Often, servers include headers that allow the client to figure out why the request could not be satisfied. See "Content Negotiation and Transcoding" in Chapter 17 for more information.
407	Proxy Authentication Required	Like the 401 status code, but used for proxy servers that require authentication for a resource.
408	Request Timeout	If a client takes too long to complete its request, a server can send back this status code and close down the connection. The length of this timeout varies from server to server but generally is long enough to accommodate any legitimate request.
409	Conflict	Used to indicate some conflict that the request may be causing on a resource. Servers might send this code when they fear that a request could cause a conflict. The response should contain a body describing the conflict.
410	Gone	Similar to 404, except that the server once held the resource. Used mostly for web site maintenance, so a server's administrator can notify clients when a resource has been removed.

Table 3-9. Client error status codes and reason phrases (continued)

Status code	Reason phrase	Meaning
411	Length Required	Used when the server requires a Content-Length header in the request message. See "Content headers" for more on the Content-Length header.
412	Precondition Failed	Used if a client makes a conditional request and one of the conditions fails. Conditional requests occur when a client includes an Expect header. See Appendix C for more on the Expect header.
413	Request Entity Too Large	Used when a client sends an entity body that is larger than the server can or wants to process.
414	Request URI Too Long	Used when a client sends a request with a request URL that is larger than the server can or wants to process.
415	Unsupported Media Type	Used when a client sends an entity of a content type that the server does not understand or support.
416	Requested Range Not Satisfiable	Used when the request message requested a range of a given resource and that range either was invalid or could not be met.
417	Expectation Failed	Used when the request contained an expectation in the Expect request header that the server could not satisfy. See Appendix C for more on the Expect header. A proxy or other intermediary application can send this response code if it has unambiguous evidence that the origin server will generate a failed expectation for the request.

500-599:SERVER HATA KODLARI

Bazen istemci dogru istegi yollar,ama serverin kendisi hatalidir.Bu istemcinin serverin limitlerini asmasi durumunda veya serverin alt bilesenlerinde hata olmasi durumunda gerceklesebilir,mesela gateway kaynagi gibi.

Proxyler genelde problemleri serverlarla istemci adina konusarak ortadan kaldirir.Proxyler 5XX server hata kodlarini tanimlar.(Bolum 6da konuscaz Proxyler Hakkinda).Tablo buyrun.

Table 3-10. Server error status codes and reason phrases

Status code	Reason phrase	Meaning
500	Internal Server Error	Used when the server encounters an error that prevents it from servicing the request.
501	Not Implemented	Used when a client makes a request that is beyond the server's capabilities (e.g., using a request method that the server does not support).
502	Bad Gateway	Used when a server acting as a proxy or gateway encounters a bogus response from the next link in the request response chain (e.g., if it is unable to connect to its parent gateway).
503	Service Unavailable	Used to indicate that the server currently cannot service the request but will be able to in the future. If the server knows when the resource will become available, it can include a Retry-After header in the response. See "Response Headers" for more on the Retry-After header.

Table 3-10. Server error status codes and reason phrases (continued)

Status code	Reason phrase	Meaning
504	Gateway Timeout	Similar to status code 408, except that the response is coming from a gateway or proxy that has timed out waiting for a response to its request from another server.
505	HTTP Version Not Supported	Used when a server receives a request in a version of the protocol that it can't or won't support. Some server applications elect not to support older versions of the protocol.

HEADERLAR

Headerlar ve methodlar beraber calisilarki istemcinin ve serverin ne yaptigini aciklasinlar.Bu bolum hemen kisaca HTTP headerlarinin amaclarini ve bazi HTTP/1.1 'de tanimlanmamis headerlari tanimlicak.Appendix C bolumunde kitabin sonunda daha detayli aciklamalari bulabilirsiniz.

Her belirli tip mesaja karsili headerlar vardir,genel amacari,istek ve cevap mesajarlinin ikisinide bilgi saglamak.Headerlar 5 sekilde yer alirlar.

GENEL HEADERLAR

Bu genel headerlar istemci ve server icinde kullanilir,Genel amaclara hizmet ederler.Istemci,server ve diger uygulamalar icin kullanislidir.Ornegin,Date headeri genel amaclidir.Iki tarafa saat ve gunu verir.

Date: Tue, 3 Oct 1974 02:16:00 GMT

ISTEK HEADERLARI

Basliginda dedigi gibi,istek mesajlari icin tanimlanan headerlar.Serverlara ekstra bilgi saglarlar.Misal istemci ne tur bir data almak istiyor.Ornegin,Accept headeri servera istemcinin Kabul edecegi medya tiplerini soyler.

Accept: */*

CEVAP HEADERLARI

Cevap headerlari istemciye bilgi saglayan kendi tasarlanmis headerlari sahiptir(misal,istemci ne tur bir serverla konusuyor)Ornegin,Server headeri istemciye Version 1.0 Tiki-Hut server ile konusuyorsun diyor.

Server: Tiki-HUT/1.0

BODY HEADERLARI

Body headerlari bildiginiz gibi body ile iliskilidir.Ornegin,body'deki data tipini bize soyleyebilir ilk bolumde MIME tipinden bahsetmistik o bu iste.Ornegin,Content-type headeri yani Icerik-tipi bakalim hemen:

Content-type:text\html; charset=iso-latin-1

GENISLETILMIS HEADERLAR

Genisletilmis headerlar standartlasmis headerlar arasina girmez uygulamalar icin yazilmis headerlardir.HTTP uygulumalari toleransli olmali ve headerlari yollamali,headerlarin ne anlama geldigini bilmese bile.

GENEL HEADERLAR

Bazi headerlar basit bilgiler saglar bize.Bu headerlar genel headerlar olarak bilinir.Engellere takilmadan mesaj hakkinda ne olursa olsun bize bilgi saglar.

Ornegin,istek mesajida cevap mesajida olustursaniz,tarih ve saat mesajlarda ayni anlama gelmeli,headerda bunu saglar,bu tarz headelar iki mesaj tipi icinde ayni bilgileri saglar.Tablo geliyor simdi.

Table 3-11. General informational headers

Header	Description
Connection	Allows clients and servers to specify options about the request/response connection
Date[a]	Provides a date and time stamp telling when the message was created
MIME-Version	Gives the version of MIME that the sender is using
Trailer	Lists the set of headers that are in the trailer of a message encoded with the chunked transfer encoding[b]
Transfer-Encoding	Tells the receiver what encoding was performed on the message in order for it to be transported safely
Upgrade	Gives a new version or protocol that the sender would like to "upgrade" to using
Via	Shows what intermediaries (proxies, gateways) the message has gone through

[a] Appendix C lists the acceptable date formats for the Date header.
[b] Chunked transfer codings are discussed further in "Chunking and persistent connections" in Chapter 15.

GENEL ONBELLEK HEADERLARI

HTTP/1.0 cache headerlari ilk olarak bize gosteren HTTP versiyonu,ana serverdan cekmek yerine isletim sistemlerindeki gibi onbellege kopyaliniyor objeler ve daha hizli olan onbellekten aliniyor.HTTP'nin son versiyonunda zengin onbellek paramtetreleri geldi.Bolum 7 de onbelleklere bakicaz simdi basit onbellek headerlari geliyor.Tablo geldi zevke devam:D.

ISTEK HEADERLARI

Istek headerlari mantigin cagirdigi gibi istek mesajlarinda bulunur.Bize istegin kimin yolladigini yada ne yollandigini,istegin kokeni neresi veya istemci kapasitesini ve sartlari ne bunlari soyler.Server cevap verirken istemciyi daha iyi tanimak icin bu bilgileri kullanir.

Table 3-13. Request informational headers

Header	Description
Client-IP[a]	Provides the IP address of the machine on which the client is running
From	Provides the email address of the client's user[b]
Host	Gives the hostname and port of the server to which the request is being sent
Referer	Provides the URL of the document that contains the current request URI
UA-Color	Provides information about the color capabilities of the client machine's display
UA-CPU[c]	Gives the type or manufacturer of the client's CPU
UA-Disp	Provides information about the client's display (screen) capabilities
UA-OS	Gives the name and version of operating system running on the client machine
UA-Pixels	Provides pixel information about the client machine's display
User-Agent	Tells the server the name of the application making the request

[a] Client-IP and the UA-* headers are not defined in RFC 2616 but are implemented by many HTTP client applications.
[b] An RFC 822 email address format.
[c] While implemented by some clients, the UA-* headers can be considered harmful. Content, specifically HTML, should not be targeted at specific client configurations.

ACCEPT HEADERLARI

Accept headerlari servera istemcinin kapasitesini ve sartlarini soyler,ne istiyorlar,ne kullanabilirler,ve,en onemlisi,ne istemiyorlar.Serverlar bu ekstra bilgileri daha akillica kararlar vermek icin kullanirlar.Accept headeri iki tarafin iletisiminde yararlidir.Istemci ne istedigini soyler,ve serverlar istemcinin zamanini bosa harcamaz ve bantaraliginda istemcinin kullanamayacagi seyleri yollamaz.Tablo geliyor.

Table 3-14. Accept headers

Header	Description
Accept	Tells the server what media types are okay to send
Accept-Charset	Tells the server what charsets are okay to send
Accept-Encoding	Tells the server what encodings are okay to send
Accept-Language	Tells the server what languages are okay to send
TE[a]	Tells the server what extension transfer codings are okay to use

[a] See "Transfer-Encoding Headers" in Chapter 15 for more on the TE header.

ISTEGE BAGLI ISTEK HEADERLARI

Bazen,istemci isteginde bazi kisitlamalara gidebilir.Ornegin,eger istemci documentin kopyasina sahipse,servera kendi elinde bulunan kopyadan baska bir kopya varsa onu gonder diyebilir.Istek headerlarinda bu tarz sartlar kullanabilir,istemci istege kisitlama koyabilir.Server istegi tamamlamadan once kisitlamalarin dogru olduguna bakmali.Asagiadaki tabloda cesitli sart istek headerlari var.

ISTEK GUVENLIK HEADERLARI

HTTP dogal olarak istekler icin cevaplarda kimlik dogrulayarak guvenligi saglamaya gider.Istemcinin kaynaga ulasmadan once daha guvenli islem yapilmasi icin kimlik kontrolune gider.Bolum 14'te tartisicaz.HTTP'nin bunu yapmaya calistigini bilin yani.Simdi diger guvenlik headerlari ile beraber geliyor.

Table 3-16. Request security headers

Header	Description
Authorization	Contains the data the client is supplying to the server to authenticate itself
Cookie	Used by clients to pass a token to the server—not a true security header, but it does have security implications[a]
Cookie2	Used to note the version of cookies a requestor supports; see "Version 1 (RFC 2965) Cookies" in Chapter 11

[a] The Cookie header is not defined in RFC 2616; it is discussed in detail in Chapter 11.

PROXY ISTEK HEADERLARI

Proxyler su anki Internette baya bir artis icinde,ve fonksyionlari daha iyi yerine getirmesi icin birkac header var.Bolum 6'da proxylere bakicaz.Bolum 6 gelsin artik sabahtan beri bolum 6 biktim kardesim yani.Tablo geliyor.

Table 3-17. Proxy request headers

Header	Description
Max-Forwards	The maximum number of times a request should be forwarded to another proxy or gateway on its way to the origin server—used with the TRACE method[a]
Proxy-Authorization	Same as Authorization, but used when authenticating with a proxy
Proxy-Connection	Same as Connection, but used when establishing connections with a proxy

[a] See "Max-Forwards" in Chapter 6.

CEVAP HEADERLARI

Cevap mesajlari kendilerine ait cevap headerlarina sahipler.Cevap headerlari istemciye ekstra bilgi saglar.Cevabi kim yolluyor,cevabi verenin kapasitesi,cevap hakkinda ozel yonlendirlemeleri bile cevap kisminda belirtir.Bu headerlar istemciye cevapla daha iyi anlasmasini ve gelecekte daha iyi istekler yapmasini saglar.Tablo Geliyor.

Table 3-18. Response informational headers

Header	Description
Age	How old the response is[a]
Public[b]	A list of request methods the server supports for its resources
Retry-After	A date or time to try back, if a resource is unavailable
Server	The name and version of the server's application software
Title[c]	For HTML documents, the title as given by the HTML document source
Warning	A more detailed warning message than what is in the reason phrase

[a] Implies that the response has traveled through an intermediary, possibly from a proxy cache.
[b] The Public header is defined in RFC 2068 but does not appear in the latest HTTP definition (RFC 2616).
[c] The Title header is not defined in RFC 2616; see the original HTTP/1.0 draft definition (*http://www.w3.org/Protocols/HTTP/HTTP2.html*).

ANLASMA HEADERLARI

HTTP/1.1 serverlara ve istemcilere eger birden fazla gosterim varsa kaynakla anlasma saglar.Ornegin,bir documentte hem Almanca hem de Fransizca cevirmesi olmasi.Bolum 17 bunun uzerinden yurucek.Burda simdi birkac header var kaynakla anlasmak icin.Tablooooo.

Table 3-19. Negotiation headers

Header	Description
Accept-Ranges	The type of ranges that a server will accept for this resource
Vary	A list of other headers that the server looks at and that may cause the response to vary; i.e., a list of headers the server looks at to pick which is the best version of a resource to send the client

CEVAP GUVENLIK HEADERLARI

Istek guvenlik headerlarini gorduk,Cevap tarafindada kimlik dogrulama olaylari var.Bolum 14 de konusucaz,Simdilik,basit headerlari gorucez.Tablo geliyor.

Table 3-20. Response security headers

Header	Description
Proxy-Authenticate	A list of challenges for the client from the proxy
Set-Cookie	Not a true security header, but it has security implications; used to set a token on the client side that the server can use to identify the client[a]
Set-Cookie2	Similar to Set-Cookie, RFC 2965 Cookie definition; see "Version 1 (RFC 2965) Cookies" in Chapter 11
WWW-Authenticate	A list of challenges for the client from the server

[a] Set-Cookie and Set-Cookie2 are extension headers that are also covered in Chapter 11.

BODY HEADERLARI

HTTP mesajlarinin icerigini tanimlayan bir suru header var.Cunku hem istek hemde cevap mesajlari body icerebilir.Bu headerlar iki mesaj tipindede gorulur.

Body headerlari icerik hakkinda genis bilgi yelpazesi saglar.Genelde,body headerlari aliciya mesajin ne ile ugrastigini soyler.

Table 3-21. Entity informational headers

Header	Description
Allow	Lists the request methods that can be performed on this entity
Location	Tells the client where the entity really is located; used in directing the receiver to a (possibly new) location (URL) for the resource

ICERIK HEADERLARI

Icerik headerlari body icerigi hakkinda belirgin bilgiler sagar,tip aciklamasi,boyutu,ve kullanisli diger bilgiler.Ornek,bir web tarayicisi content-type headerina bakar objeyi nasil gosterecegini bilir.Content(Icerik)-headerlari geliyor.

Table 3-22. Content headers

Header	Description
Content-Base[a]	The base URL for resolving relative URLs within the body
Content-Encoding	Any encoding that was performed on the body

Table 3-22. Content headers (continued)

Header	Description
Content-Language	The natural language that is best used to understand the body
Content-Length	The length or size of the body
Content-Location	Where the resource actually is located
Content-MD5	An MD5 checksum of the body
Content-Range	The range of bytes that this entity represents from the entire resource
Content-Type	The type of object that this body is

[a] The Content-Base header is not defined in RFC 2616.

BODY ONBELLEK HEADERLARI

Bu genel onbellek headerlari ne zaman ve nasil onbellege alinacagini saglar.Body onbellek headerlari ise onbellege alinan varlik hakkinda bilgi barindirir.Ornegin,onbellege kopyalanmis kaynak hakkinda bilgiye ihtiyac var ve artik onbellekte bulunmayacak kaynagi nasil degerlendirebiliriz diye bir ipucu verebilir.

Bolum 7 de derin bir sekilde HTTP istek ve cevaplari incelenecek.Headerlari gene gorucez.Tablo geliyor.

Table 3-23. Entity caching headers

Header	Description
ETag	The entity tag associated with this entity[a]
Expires	The date and time at which this entity will no longer be valid and will need to be fetched from the original source
Last-Modified	The last date and time when this entity changed

[a] Entity tags are basically identifiers for a particular version of a resource.