

Chapter - 14

HTTP क्या है?

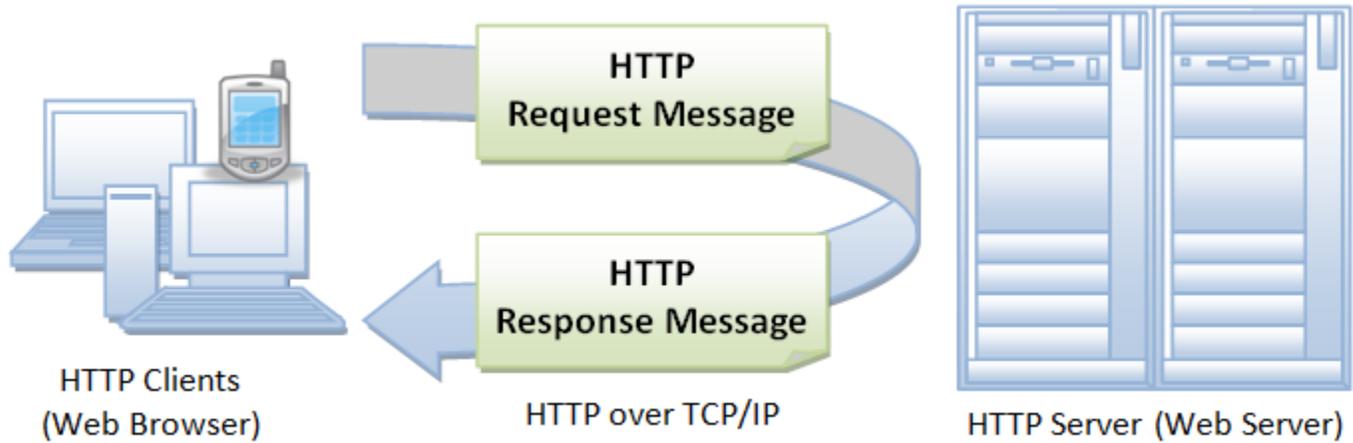
HTTP क्या है? (What is HTTP)

HTTP का पूरा नाम हाइपरटेक्स्ट ट्रांसफर प्रोटोकॉल है जिसका उपयोग वेब पर डेटा ट्रांसफर करने के लिए किया जाता है। यह इंटरनेट प्रोटोकॉल सूट का हिस्सा है और वेबपेज डेटा ट्रांसमिट करने के लिए इस्तेमाल की जाने वाली कमांड और सेवाओं को परिभाषित करता है। एड्रेस के सामने `http://` इंटर करने वाले किसी भी वेब पेज तक पहुंचने पर ब्राउज़र को HTTP पर कन्नेक्ट करने के लिए कहता है। उदाहरण के लिए, कंप्यूटर हिंदी नोट्स का URL `https://www.computerhindinotes.com` है। आज के ब्राउज़र्स को अब URL के सामने HTTP की आवश्यकता नहीं है क्योंकि यह संचार का डिफ़ॉल्ट तरीका है। हालाँकि, इसे अलग-अलग प्रोटोकॉल जैसे FTP की आवश्यकता के कारण ब्राउज़र्स में रखा जाता है।

HTTP एक सर्वर-क्लाइंट मॉडल का उपयोग करता है। एक क्लाइंट उदाहरण के लिए, होम कंप्यूटर, लैपटॉप या मोबाइल डिवाइस हो सकता है। HTTP सर्वर आमतौर पर वेब सर्वर पर चलने वाला वेब होस्ट है, जैसे Apache या IIS। जब आप एक वेबसाइट का उपयोग करते हैं, तो आपका ब्राउज़र संबंधित वेब सर्वर को एक रिक्वेस्ट भेजता है और यह HTTP स्थिति कोड के साथ रिस्पॉन्स करता है। यदि URL वैध है और कनेक्शन दिया गया है, तो सर्वर आपके ब्राउज़र को वेबपेज और संबंधित फ़ाइलों को भेज देगा।

यह इंटरनेट पर सूचनाओं को ट्रांसफर करने का एक प्रोटोकॉल है। इसी प्रोटोकॉल के उपयोग ने बाद में वर्ल्ड वाइड वेब को जन्म दिया था। इस प्रोटोकॉल का विकास वर्ल्ड वाइड वेब कॉन्सोर्टियम तथा इंटरनेट इंजीनियरिंग टास्कफोर्स ने सम्मिलित रूप से किया था।

यह एक स्टैंडर्ड इंटरनेट प्रोटोकॉल है । यह वेब ब्राउजर जैसे माइक्रोसॉफ्ट इंटरनेट एक्सप्लोरर और वेब सर्वर जैसे माइक्रोसॉफ्ट इंटरनेट इनफॉर्मेशन सर्विसेस (IIS) के बीच क्लाइंट/सर्वर इंटरैक्शन प्रोसेस (आदान-प्रदान की प्रक्रिया) को स्पेसीफाय (वर्णित) करता है।



वास्तविक हाईपरटेक्स्ट ट्रान्सफर प्रोटोकॉल 1.0 एक स्टेटलेस प्रोटोकॉल है, जिसके द्वारा वेब ब्राउजर वेब सर्वर से कनेक्शन (जुड़ता) करता है व उपयुक्त फाइल को डाउनलोड करता है और फिर कनेक्शन खत्म करता है। ब्राउजर सामान्यतया एक फाइल के उपयोग के लिए HTTP से रिक्वेस्ट करता है। GET मेथड TCP पोर्ट 80 पर रिक्वेस्ट करती है, जिसमें HTTP रिक्वेस्ट हेडर की सीरीज होती है, जो ट्रांजेक्शन मेथड (GET, POST, HEAD) इत्यादि को परिभाषित करती है और साथ ही सर्वर को क्लाइंट की क्षमता को बताती है।

सर्वर HTTP रिस्पॉन्स हेडर की सीरीज को रिस्पॉन्स देता है, जो दर्शाता है कि ट्रांजेक्शन सफल रहा है कि नहीं, किस प्रकार डेटा भेजा गया है, सर्वर का प्रकार और जो डेटा भेजा गया था, इत्यादि IIS 4 प्रोटोकॉल के उस नए वर्जन को सपोर्ट करता है, जिसे HTTP 1.1 कहा गया। नए गुणों के कारण यह ज्यादा सक्षम है।

HTTPS क्या है? (What is HTTPS?)

HTTPS का पूरा नाम Hypertext Transfer Protocol Secure है यह एक प्रोटोकॉल है जो ट्रांसपोर्ट-लेयर सिक्योरिटी द्वारा एन्क्रिप्ट किए गए कनेक्शन पर HTTP का उपयोग करता है। HTTPS का उपयोग संचारित डेटा (transmitted data) को ईव्सड्रॉपिंग (eavesdropping) से बचाने के लिए किया जाता है। यह वेब का डिफॉल्ट प्रोटोकॉल है, और वेबसाइट के यूजर को सरकार या ISP द्वारा सेंसरशिप से सुरक्षित कर सकता है।

- HTTPS अपनी जानकारी को स्थानांतरित करने के लिए पोर्ट 443 का उपयोग करता है।
- HTTPS का उपयोग पहली बार HTTP / 1.1 में किया गया है और इसे RFC 2616 में परिभाषित किया गया है।

Features of HTTP (HTTP की विशेषताएं)

HTTP की तीन बेसिक विशेषताएं हैं जो HTTP को एक सरल और शक्तिशाली प्रोटोकॉल बनाती हैं:

HTTP कनेक्शन रहित है (HTTP is connectionless): HTTP क्लाइंट, अर्थात् ब्राउजर HTTP रिक्वेस्ट शुरू करता है और एक रिक्वेस्ट हो जाने के बाद, क्लाइंट प्रोसेस का इंतजार करता है। सर्वर रिक्वेस्ट को प्रोसेस करता है

और एक प्रतिक्रिया (Response) भेजता है जिसके बाद क्लाइंट कनेक्शन को डिस्कनेक्ट करता है। इसलिए क्लाइंट और सर्वर वर्तमान रिक्वेस्ट और प्रतिक्रिया (Response) के दौरान एक दूसरे के बारे में जानते हैं। नए कनेक्शन पर आगे रिक्वेस्ट किए जाते हैं जैसे क्लाइंट और सर्वर एक दूसरे के लिए नए हैं।

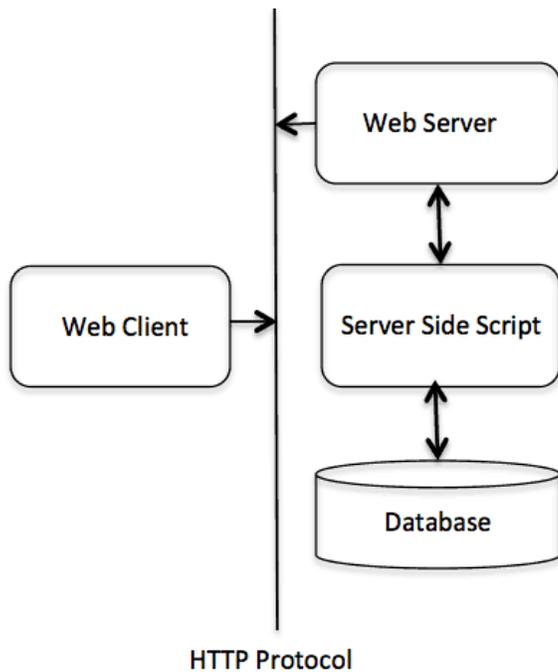
HTTP स्वतंत्र मीडिया है (HTTP is media independent): इसका मतलब है, किसी भी प्रकार का डेटा HTTP द्वारा भेजा जा सकता है जब तक कि क्लाइंट और सर्वर दोनों को पता है कि डेटा कंटेंट को कैसे संभालना है। क्लाइंट के साथ-साथ उपयुक्त MIME का उपयोग करके कंटेंट प्रकार निर्दिष्ट करने के लिए सर्वर की आवश्यकता होती है।

HTTP स्टेटलेस है (HTTP is stateless): जैसा कि ऊपर बताया गया है, HTTP कनेक्शन रहित है इसी तरह यह HTTP का एक स्टेटलेस प्रोटोकॉल है। सर्वर और क्लाइंट केवल एक वर्तमान रिक्वेस्ट के दौरान एक दूसरे के बारे में जानते हैं। बाद में, दोनों एक-दूसरे के बारे में भूल जाते हैं। प्रोटोकॉल की इस प्रकृति के कारण, न तो क्लाइंट और न ही ब्राउज़र वेब पेजों पर विभिन्न रिक्वेस्ट के बीच जानकारी को बनाए रख सकते हैं।

HTTP / 1.0 प्रत्येक request/response exchange के लिए एक नए कनेक्शन का उपयोग करता है, जहां HTTP / 1.1 कनेक्शन का उपयोग एक या अधिक request/response exchange के लिए किया जा सकता है।

Basic Architecture of HTTP (HTTP का बेसिक आर्किटेक्चर)

निम्नलिखित आरेख एक वेब एप्लिकेशन की बहुत ही मूल संरचना को दर्शाता है -



HTTP प्रोटोकॉल क्लाइंट / सर्वर आधारित आर्किटेक्चर पर आधारित एक रिक्वेस्ट / रिस्पॉन्स प्रोटोकॉल है जहां वेब ब्राउज़र, रोबोट और सर्च इंजन आदि HTTP क्लाइंट की तरह काम करते हैं और वेब सर्वर एक सर्वर की तरह काम करता है।

- **Client**

HTTP क्लाइंट एक रिक्वेस्ट मेथड, URI, और प्रोटोकॉल वर्जन के रूप में सर्वर को MIME के द्वारा एक रिक्वेस्ट भेजता है।

- **Server**

HTTP सर्वर एक Status line के साथ responds देता है, जिसमें मैसेज प्रोटोकॉल वर्जन और एक सफलता या त्रुटि कोड शामिल होता है, जिसे MIME के द्वारा पूरा किया जाता है जैसे सर्वर जानकारी, एंटीटी मेटा जानकारी और possible entity-body content शामिल होता है।

What are HTTP status codes?

HTTP स्टेटस कोड की सूची नीचे दी गई है। ये कोड error messages हैं जो क्लाइंट को किसी अन्य कंप्यूटर या डिवाइस को HTTP पर एक्सेस करने की अनुमति देता है ताकि यह पता चल सके कि कैसे आगे बढ़ना है या नहीं। उदाहरण के लिए, 404 error ब्राउज़र को बताता है कि सर्वर पर रिक्वेस्ट मौजूद नहीं है।\

1xx – 2xx	3xx – 4xx	5xx
100 (Continue)	301 (Moved permanently)	500 (Internal server error)
101 (Switch protocols)	302 (Moved temporarily)	501 (Not Implemented)
102 (Processing)	304 (Loaded Cached copy)	502 (Bad gateway)
200 (Success)	307 (Internal redirect)	503 (Service unavailable)
201 (Fulfilled)	400 (Bad request)	504 (Gateway timeout)
202 (Accepted)	401 (Authorization required)	505 (HTTP version not supported)
204 (No content)	402 (Payment required)	506 (Variant also negotiates)
205 (Reset content)	403 (Forbidden)	507 (Insufficient storage)
206 (Partial content)	404 (Not found)	510 (Not extended)
207 (Multi-Status)	405 (Method not allowed)	
	406 (Not acceptable)	
	407 (Proxy authentication required)	

	408 (Request timeout)	
	409 (Conflict)	
	410 (Gone)	
	411 (Length required)	
	412 (Precondition failed)	
	413 (Request entity too large)	
	414 (Request URI too large)	
	415 (Unsupported media type)	
	416 (Request range not satisfiable)	
	417 (Expectation failed)	
	422 (Unprocessable entity)	
	423 (Locked)	