WebSocket

[Naar navigatie springen](https://nl.wikipedia.org/wiki/WebSocket#mw-head)[Naar zoeken springen](https://nl.wikipedia.org/wiki/WebSocket#searchInput)

**WebSocket** is een [netwerkprotocol](https://nl.wikipedia.org/wiki/Netwerkprotocol) dat [full-duplex](https://nl.wikipedia.org/wiki/Full-duplex) communicatie biedt over een enkele [TCP](https://nl.wikipedia.org/wiki/Transmission_Control_Protocol)-verbinding. Het WebSocketprotocol is gestandaardiseerd door de [Internet Engineering Task Force](https://nl.wikipedia.org/wiki/Internet_Engineering_Task_Force) in 2011 als [RFC](https://nl.wikipedia.org/wiki/Request_for_Comments) [6455](https://datatracker.ietf.org/doc/html/rfc6455) en de WebSocketAPI gemaakt in [Web IDL](https://nl.wikipedia.org/w/index.php?title=Web_IDL&action=edit&redlink=1) is gestandaardiseerd door het [World Wide Web Consortium](https://nl.wikipedia.org/wiki/World_Wide_Web_Consortium).

WebSocket is ontworpen om te worden toegepast in [webbrowsers](https://nl.wikipedia.org/wiki/Webbrowser) en [webservers](https://nl.wikipedia.org/wiki/Webserver). Het WebSocketprotocol is een zelfstandig op [TCP](https://nl.wikipedia.org/wiki/Transmission_Control_Protocol) gebaseerd protocol. WebSocket vertoont enige gelijkenis met [HTTP](https://nl.wikipedia.org/wiki/Hypertext_Transfer_Protocol), omdat de [handshake](https://nl.wikipedia.org/w/index.php?title=Handshake&action=edit&redlink=1" \o "Handshake (de pagina bestaat niet)) van het WebSocketprotocol door [HTTP](https://nl.wikipedia.org/wiki/Hypertext_Transfer_Protocol)-servers wordt geïnterpreteerd als een Upgraderequest. Het WebSocketprotocol maakt bidirectioneel dataverkeer mogelijk tussen webbrowsers en webservers. Dit kan doordat het protocol zorgt voor een standaard manier van dataverkeer zonder de noodzaak van een request door de webbrowser om vanaf de server data te sturen naar de browser. Daarnaast staat het protocol toe berichten heen en weer te sturen en daarbij de connectie tussentijds open te houden. Op deze manier is er een voortdurende bidirectionele conversatie tussen webserver en webbrowser. De communicatie vindt plaats over TCP-poort 80 (of TCP-poort 443 als de communicatie over TLS-versleutelde connecties gaat), wat een voordeel is in netwerkomgevingen waar poorten die niet bedoeld zijn voor het web (andere TCP-poorten dan 80 en 443) worden geblokkeerd door een [firewall](https://nl.wikipedia.org/wiki/Firewall). Soortgelijk bidirectioneel dataverkeer is op andere niet-gestandaardiseerde manieren tot stand gebracht met stopgaptechnologieën zoals [Comet](https://nl.wikipedia.org/wiki/Comet_(internet)" \o "Comet (internet)).

Het WebSocketprotocol wordt momenteel ondersteund in de meeste grote browsers, waaronder: [Google Chrome](https://nl.wikipedia.org/wiki/Google_Chrome), [Microsoft Edge](https://nl.wikipedia.org/wiki/Microsoft_Edge), [Internet Explorer](https://nl.wikipedia.org/wiki/Internet_Explorer), [Firefox](https://nl.wikipedia.org/wiki/Mozilla_Firefox), [Safari](https://nl.wikipedia.org/wiki/Safari_(webbrowser)) en [Opera](https://nl.wikipedia.org/wiki/Opera_(webbrowser)). WebSocket heeft ook [webapplicaties](https://nl.wikipedia.org/wiki/Webapplicatie) op de server nodig om het te ondersteunen.



**Inhoud**

* [1Overzicht](https://nl.wikipedia.org/wiki/WebSocket#Overzicht)
* [2Geschiedenis](https://nl.wikipedia.org/wiki/WebSocket#Geschiedenis)
* [3Protocolhandshake](https://nl.wikipedia.org/wiki/WebSocket#Protocolhandshake)

Overzicht[[bewerken](https://nl.wikipedia.org/w/index.php?title=WebSocket&veaction=edit&section=1" \o "Bewerk dit kopje: Overzicht) | [brontekst bewerken](https://nl.wikipedia.org/w/index.php?title=WebSocket&action=edit&section=1)]

In tegenstelling tot HTTP biedt WebSocket full-duplex-communicatie. Verder zorgt WebSocket voor de mogelijkheid een stroom van berichten te sturen bovenop een TCP-verbinding. TCP zelf regelt alleen een stroom van bytes en kent geen berichtenconcept. Voor WebSocket was full-duplexcommunicatie over poort 80 mogelijk met Comet-channels. Echter, de Comet-implementatie was geen standaard. Ook was de Comet-implementatie zwaarder omdat het werkte bovenop HTTP en het was inefficiënt voor kleine berichten. Het WebSocketprotocol heeft tot doel deze problemen te overwinnen zonder de veiligheid van het web in het geding te laten komen.

De WebSocketspecificatie definieert ws en wss als twee nieuwe [URI](https://nl.wikipedia.org/wiki/Uniform_Resource_Identifier)-schema's, welke gebruikt kunnen worden voor niet-versleutelde en versleutelde verbindingen respectievelijk. Buiten fragmenten en schemanaam wordt de rest van de [URI](https://nl.wikipedia.org/wiki/Uniform_Resource_Identifier)-componenten gedefinieerd volgens [URI generic syntax](https://nl.wikipedia.org/wiki/Uniform_Resource_Identifier).

Met gebruik van de Development Tools van de browser kunnen ontwikkelaars WebSockethandshakes en WebSocketframes inspecteren.

Geschiedenis[[bewerken](https://nl.wikipedia.org/w/index.php?title=WebSocket&veaction=edit&section=2" \o "Bewerk dit kopje: Geschiedenis) | [brontekst bewerken](https://nl.wikipedia.org/w/index.php?title=WebSocket&action=edit&section=2)]

WebSocket werd eerst [TCP](https://nl.wikipedia.org/wiki/TCP/IP" \o "TCP/IP)Connection genoemd in de [HTML5](https://nl.wikipedia.org/wiki/HTML5)-specificatie als vervanger voor een op [TCP](https://nl.wikipedia.org/wiki/TCP/IP) gebaseerde socket-[API](https://nl.wikipedia.org/wiki/Application_programming_interface). In juni 2008 werden discussies geleid door Michael Carter, welke resulteerden in de eerste versie van het nu hetende WebSocket.

Na deze discussies is door samenwerking tussen Ian Hickson en Michael Carter in de #whatwg [IRC](https://nl.wikipedia.org/wiki/Internet_Relay_Chat)-[chatroom](https://nl.wikipedia.org/wiki/Chatroom) de naam WebSocket bedacht. Daarop volgde toevoeging aan de [HTML5](https://nl.wikipedia.org/wiki/HTML5)-specificatie door Ian Hickson en werd de toevoeging aangekondigd op de [cometdaily](https://nl.wikipedia.org/w/index.php?title=Cometdaily&action=edit&redlink=1" \o "Cometdaily (de pagina bestaat niet))-[blog](https://nl.wikipedia.org/wiki/Weblog) van Michael Carter. In december 2009 was [Google Chrome 4](https://nl.wikipedia.org/wiki/Google_Chrome) de eerste grote webbrowser die de volledige standaard met WebSocket standaard aangezet ondersteunde. De ontwikkeling van het WebSocketprotocol werd daarna van het [W3C](https://nl.wikipedia.org/wiki/World_Wide_Web_Consortium) en de whatwg-groep verplaatst naar het [IETF](https://nl.wikipedia.org/wiki/Internet_Engineering_Task_Force) in februari 2010 met twee revisies onder controle van Ian Hickson.

Nadat het protocol werd ondersteund door meerdere browsers en standaard aangezet was, is de [RFC](https://nl.wikipedia.org/wiki/Request_for_Comments) afgemaakt onder Ian Fette in december 2011.

Protocolhandshake[[bewerken](https://nl.wikipedia.org/w/index.php?title=WebSocket&veaction=edit&section=3" \o "Bewerk dit kopje: Protocolhandshake) | [brontekst bewerken](https://nl.wikipedia.org/w/index.php?title=WebSocket&action=edit&section=3)]

Om een WebSocketconnectie te starten, stuurt de browser een WebSockethandshakerequest naar de server. De server reageert hierop met antwoord, zoals weergegeven is in de onderstaande figuur.

WebSockethandshakerequest (net zoals bij HTTP, elke regel moet eindigen met [\r](https://nl.wikipedia.org/wiki/Carriage_return)[\n](https://nl.wikipedia.org/wiki/Line_feed) en er moet een lege regel aan het einde overblijven)

GET **/chat** **HTTP**/1.1

Host: server.example.com

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==

Sec-WebSocket-Protocol: chat, superchat

Sec-WebSocket-Version: 13

Origin: http://example.com

Serverantwoord:

**HTTP**/1.1 101 **Switching Protocols**

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Accept: HSmrc0sMlYUkAGmm5OPpG2HaGWk=

Sec-WebSocket-Protocol: chat

De WebSockethandshake lijkt op een [HTTP-request](https://nl.wikipedia.org/wiki/Hypertext_Transfer_Protocol), zodat webservers zowel HTTP-connecties als WebSocketconnecties kunnen verwerken op dezelfde poort. Zodra de connectie is gemaakt door middel van de WebSockethandshake, wisselt de communicatie naar een [bidirectioneel](https://nl.wikipedia.org/wiki/Full-duplex" \o "Full-duplex) [binair](https://nl.wikipedia.org/wiki/Binair) protocol, die niet meer voldoet aan het HTTP-protocol.

Naast de upgrade-headers stuurt de webbrowser ook de Sec-WebSocket-Key-[header](https://nl.wikipedia.org/wiki/Hoofding) mee. Deze header bevat [base64](https://nl.wikipedia.org/wiki/Base64)-gecodeerde willekeurige bytes die fungeren als een sleutel, de server reageert hierop met een [hash](https://nl.wikipedia.org/wiki/Hashfunctie" \o "Hashfunctie) van de sleutel in de Sec-WebSocket-Accept-header. Dit wordt gedaan om te voorkomen dat een [cache](https://nl.wikipedia.org/wiki/Cache_(tijdelijk_geheugen)) [proxy](https://nl.wikipedia.org/wiki/Proxyserver) een eerder verzonden conversatie opnieuw verstuurt. Dit mechanisme zorgt niet voor enige vorm van [privacy](https://nl.wikipedia.org/wiki/Privacy), [authenticatie](https://nl.wikipedia.org/wiki/Authenticatie) of integriteit. De functie die de [hash](https://nl.wikipedia.org/wiki/Hashfunctie" \o "Hashfunctie) maakt, voegt de volgende tekst toe: 258EAFA5-E914-47DA-95CA-C5AB0DC85B11(de [Universally unique identifier](https://nl.wikipedia.org/wiki/Universally_unique_identifier" \o "Universally unique identifier) van de webserver) aan de inhoud van Sec-WebSocket-Key-header (zonder de Sec-WebSocket-Key-header te decoderen vanuit base64), voert hier dan de [SHA-1](https://nl.wikipedia.org/wiki/SHA-1)-[hashingfunctie](https://nl.wikipedia.org/wiki/Hashfunctie) op uit en het resultaat wordt gecodeerd met [base64](https://nl.wikipedia.org/wiki/Base64).

Wanneer de connectie gemaakt is, kunnen de webbrowser en server data- of text-frames heen en weer sturen in [full-duplex](https://nl.wikipedia.org/wiki/Full-duplex)-modus. De data wordt voorafgegaan aan een kleine header, welke gevolgd wordt door de ladingdata. WebSocket-transmissies worden omschreven als "messages", waarbij elke message kan worden verdeeld over meerdere text-frames. Dit biedt de mogelijkheid messages te sturen waarbij de totale lengte onbekend is maar de eerste data al wel beschikbaar is (het stuurt het ene data-frame na het andere totdat het einde bereikt is en wordt gemarkeerd met de FIN bit). Met uitbreidingen van het protocol kan dit ook worden gebruikt om meerdere datastromen simultaan te versturen en ontvangen, ook wel [multiplexing](https://nl.wikipedia.org/wiki/Multiplexing" \o "Multiplexing) genoemd.

Vanuit een veiligheidsoogpunt is het belangrijk om de Origin-header te controleren tijdens het maken van een connectie vanuit de server. Zo voorkomt men Cross-Site WebSocket Hijacking-attacks, welke mogelijk zijn als de connectie wordt gecontroleerd met [cookies](https://nl.wikipedia.org/wiki/Cookie_(internet)) of HTTP-authenticatie. Het is beter sleutels of vergelijkbare mechanismen te gebruiken om de betrouwbaarheid van een WebSocketconnectie te waarborgen wanneer er gevoelige informatie wordt verstuurd over de verbinding.