技術筆記

筆記更新紀錄

|  |  |  |
| --- | --- | --- |
| 日期 yyyy/MM/dd | 更新人員 | 更新備註 |
| 2020/01/19 | Miles | 建立文件 |
| 2020/01/20 | Miles | 更新HTTP相關解說，添加TCP/IP相關解說 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

目錄

[1 了解HTTP的基本簡介 4](#_Toc30435094)

[2 了解HTTP的基本運作流程 5](#_Toc30435095)

[3. HTTP傳輸格式 7](#_Toc30435096)

[4 持續連線 9](#_Toc30435097)

[5 關於應用 10](#_Toc30435098)

[6 何謂TCP/IP(1) 11](#_Toc30435099)

[7 何謂TCP/IP(2) 11](#_Toc30435100)

[8 何謂Socket 11](#_Toc30435101)

[9 了解HTTPS的基本簡介 12](#_Toc30435102)

[10 13](#_Toc30435103)

[11 13](#_Toc30435104)

[12 13](#_Toc30435105)

## 1 了解HTTP的基本簡介

|  |
| --- |
| HTTP是什麼? |
| HTTP全名!? |
| 超本文傳輸協定(HyperText Transfer Protocol)。 |
| HTTP 由誰或者是由哪間公司提出!? |
| HTTP的發展是由提姆·柏內茲-李於1989年在歐洲核子研究組織所發起。 |
| HTTP 是做什麼的? |
| HTTP 是一種傳輸通訊協定，可以進行HTML文檔、數據資源、圖片、影片、腳本等數據的傳輸協定。  主要被設計來讓瀏覽器與伺服器進行溝通。 |
| 參考網站 |
| 1. <https://vimsky.com/zh-tw/examples/detail/java-class-java.security.interfaces.RSAPrivateCrtKey.html>(中文) 2. <https://segmentfault.com/a/1190000012368732>(中文) 3. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>(英文) |

## 2 了解HTTP的基本運作流程

|  |
| --- |
| HTTP的運作方式 |
| 基本流程 |
| 首先客戶端(瀏覽器)與伺服端(Server)傳輸並發送請求(Request)，等伺服端接收到請求之後進行一連串後續的處理之後，將資訊經由HTTP傳輸協定再次與瀏覽器傳輸並且將回應(Response)傳送回去，並且中斷連線。 |
| 基本用詞解釋 |
| 1. 傳輸動作是使用TCP的協定。 2. TCP則位於OSI模組的傳輸層。(OSI模組與TCP參考後續講解)   2. IP則位於OSI模組的網路層。(OSI模組與IP參考後續講解)  3. HTTP 則位於OSI模組的應用層。(OSI模組參考後續講解)  4. 三者是不同的東西，各自扮演不同的角色。  5. Socket是什麼?(請參考後續講解)。 |
| 基本流程參考 |
|  |
| 基本流程2 |
| 1. 客戶端(瀏覽器)發起一個HTTP請求到伺服端上的預設埠(ㄅㄨˋ)(一般預設為80)。其中以上動作我們稱之為用戶代理程式。  2. 應答的伺服器上進行資源的存取。我們稱以上為源伺服器。  3. 而用戶代理程式與源伺服器之間可能存在許多中間層，例如代理伺服器(Proxy Server)、閘道、  隧道等等。 |
| 基本用詞解釋 |
| 1. 瀏覽器(Browser)  2. 伺服器(Server)  3. 用戶代理程式(User Agent)  4. 源伺服器(Origin Server) |
| HTTP的版本 |
| HTTP 有多個版本  1. HTTP/0.9 : 只支援GET。  2. HTTP/1.0  3. HTTP/1.1  4. HTTP/2.0  5. HTTP/3.0: 最新版本2018年發布，2019年正式適用。 |
| HTTP 在OSI模型中的第幾層!? |
| HTTP 位於應用層，位於TCP(傳述層)和IP(網路層)之上，以及表示層之下。  經由加密(TCP或TLS加密的TCP)發送至應用層(也就是HTTP自己所屬的OSI模型的應用層)，因為是一個可擴充的協議，所以可以傳輸各種型態的資訊(圖片、影片..等)。 |
| 請求方法(HTTP/1.1) |
| 1. GET  2. HAND  3. POST  4. PUT  5. DELETE  6. TRACE  7. OPTIONS  8. CONNECT |

## 3. HTTP傳輸格式

|  |
| --- |
| Message Formate |
| HTTP/1.1 的請求格式(Request Message Formate)參考 |
|  |
| HTTP-message標準格式 |
| HTTP-message =  start-line  \*(header-field CRLF)  CRLF  [message-body]  #注意: CRLF是所謂哦分行(/R) |
| 格式解釋 |
| 1. start-line : 簡稱起始行，做為訊息的開始。是請求訊息與回應訊息的內容最大的差異。  其中 請求為 requset-line、回應為response-line、狀態為start-line。  請求Request message格式:    方法(method)、空白(SP)、請求目標(request-target)、再一個空白(SP)、HTTP版本、分行(CRLF)  1.1 方法必須使用大寫  1.2 請求目標必須使用/開頭  1.3 HTTP版本需要使用大寫  回應Response message格式: |
|  |
|  |
| 參考文獻 |
| 1. <https://notfalse.net/39/http-message-format> (關於HTTP-message) |

## 4 持續連線

|  |
| --- |
| 持續連線? |
| HTTP/0.9 與 HTTP/1.0 的連線機制 |
| TCP連線在每一次請求/回應之後就會關閉連線管道    從上圖可看出 每次連後接收到回應就會關閉連線通道 |
| HTTP/1.1 |
| 在HTTP 1.1中，引入了保持連線的機制，一個連接可以重複在多個請求/回應使用。持續連線的方式可以大大減少[等待時間](https://zh.wikipedia.org/wiki/%E5%BB%B6%E8%BF%9F_(%E5%B7%A5%E7%A8%8B%E5%AD%A6))，因為在發出第一個請求後，雙方不需要重新執行[TCP交握程式](https://zh.wikipedia.org/wiki/%E6%8F%A1%E6%89%8B_(%E6%8A%80%E6%9C%AF))。 |
| 差異性 |
| HTTP 1.1還改進了HTTP 1.0的頻寬。 例如，HTTP 1.1引入了[分塊傳輸編碼](https://zh.wikipedia.org/wiki/%E5%88%86%E5%9D%97%E4%BC%A0%E8%BE%93%E7%BC%96%E7%A0%81)，以允許傳遞內容可以在持續連線上被串流傳輸而不必使用到[緩衝器](https://zh.wikipedia.org/wiki/%E7%B7%A9%E8%A1%9D%E5%99%A8)。HTTP管道允許客戶端在收到每個回應之前發送多個請求，進一步減少[用戶感受到的](https://zh.wikipedia.org/wiki/%E4%BD%BF%E7%94%A8%E8%80%85%E7%B6%93%E9%A9%97)滯後時間。協定的另一個補充是位元組服務，允許客戶端請求資源的某一部份，伺服器僅回應某資源的指明部分。 |

## 5 關於應用

|  |
| --- |
| 也不一定要使用TCP |
| 為什麼不一定要使用 |
| TCP並不是對所有的應用都適合，一些新的帶有一些內在的脆弱性的運輸層協定也被設計出來。比如，即時應用並不需要甚至無法忍受TCP的可靠傳輸機制。在這種類型的應用中，通常允許一些丟包、出錯或擁塞，而不是去校正它們。例如通常不使用TCP的應用有：即時流多媒體（如[網際網路廣播](https://zh.wikipedia.org/w/index.php?title=%E5%9B%A0%E7%89%B9%E7%BD%91%E5%B9%BF%E6%92%AD&action=edit&redlink=1)）、即時多媒體播放器和遊戲、IP電話（VoIP）等等。任何不是很需要可靠性或者是想將功能減到最少的應用可以避免使用TCP。在很多情況下，當只需要多路復用應用服務時，[用戶資料報協定](https://zh.wikipedia.org/wiki/%E7%94%A8%E6%88%B7%E6%95%B0%E6%8D%AE%E6%8A%A5%E5%8D%8F%E8%AE%AE)（UDP）可以代替TCP為應用提供服務。 |

## 6 何謂TCP/IP(1)

|  |
| --- |
| TCP是什麼? |
| TCP的全名 |
| 全名:Transmission Control Protool (傳輸控制協定) |
| OSI模組定位 |
| 傳輸層 |
|  |
|  |
|  |
|  |

## 7 何謂TCP/IP(2)

|  |
| --- |
| IP是什麼? |
| IP的解釋 |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

## 8 何謂Socket

|  |
| --- |
| Socket是什麼? |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

## 9 了解HTTPS的基本簡介

|  |
| --- |
| HTTPS是什麼? |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

## 10 何謂UDP

|  |
| --- |
| UDP是什麼? |
| 全名 |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

## 11

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

## 12

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |