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 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

目錄

[1 了解HTTP的基本簡介 4](#_Toc30335258)

[2 了解HTTP的基本運作流程 4](#_Toc30335259)

[3 5](#_Toc30335260)

[4 5](#_Toc30335261)

[1 模板 5](#_Toc30335262)

## 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。 |
| 用詞解釋 |
| 1. 傳輸動作是使用TCP的協定。(傳輸層)  2. IP則是網路層。  3. HTTP 則是應用層  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。 |
| 參考流程 |
|  |
| 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。 |
| 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 |
| 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 Format |
| HTTP/1.1 的請求格式參考 |
|  |
| 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) |

## 3 何謂TCP/IP(1)

|  |
| --- |
| TCP是什麼? |
| TCP的解釋 |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

## 4 何謂TCP/IP(2)

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

## 4 何謂Socket

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

## 3

|  |
| --- |
| HTTP是什麼? |
| HTTP全名!? |
| 1. 超本文傳輸協定(HyperText Transfer Protocol)。 |
| HTTP 由誰或者是由哪間公司提出!? |
| 1. 提姆·柏內茲-李於1989年在歐洲核子研究組織所發起 |
|  |
|  |
| 參考 |
| 1. <https://vimsky.com/zh-tw/examples/detail/java-class-java.security.interfaces.RSAPrivateCrtKey.html> 2. <https://segmentfault.com/a/1190000012368732> |

## 4

|  |
| --- |
| HTTP是什麼? |
| HTTP全名!? |
| 1. 超本文傳輸協定(HyperText Transfer Protocol)。 |
| HTTP 由誰或者是由哪間公司提出!? |
| 1. 提姆·柏內茲-李於1989年在歐洲核子研究組織所發起 |
|  |
|  |
| 參考 |
| 1. <https://vimsky.com/zh-tw/examples/detail/java-class-java.security.interfaces.RSAPrivateCrtKey.html> 2. <https://segmentfault.com/a/1190000012368732> |

## 1 模板

|  |
| --- |
| 標題 |
| 問題 |
| 1. 解答 |
| HTTP 由誰或者是由哪間公司提出!? |
| 1. 提姆·柏內茲-李於1989年在歐洲核子研究組織所發起 |
|  |
|  |
| 參考 |
| 1. <https://vimsky.com/zh-tw/examples/detail/java-class-java.security.interfaces.RSAPrivateCrtKey.html> 2. <https://segmentfault.com/a/1190000012368732> |