**Server-Sent Events整合應用**

2019/5/24 Peter Pan

* **Server-Sent Events**
* **Server Side**：Server-sent events (SSE) is a server push technology enabling a browser to receive automatic updates from a server via HTTP connection. (HTTP/2)
* **Client Side**：The Server-Sent Events EventSource API is standardized as part of HTML5 by the W3C.
* **JavaEE**：Jersey has a full implementation of JAX-RS support for Server Sent Events as defined in JSR-370
* **Server-Sent Events vs. WebSockets**
* 需求面：雖然WebSocket提估完整的雙向溝通功能，很適合像遊戲、訊息交談軟體。但也有些情況必不常需要雙向溝通功能，而是單向資訊提供，例如股票報價、即時新聞等。即使偶而需要反向傳送資訊仍可借助原有的技術輕易達成 (如XMLHttpRequest)。
* SSE 使用HTTP協定原Web Server 皆已支援，不需像WebSocket需額外支援專屬協定。
* SSE 提供automatic reconnection, event ID及可發送各種自訂event 類型等功能。
* SSE 使用的EventSourceAPI 即使在不支援的瀏覽器 (如IE)，也可完全以JavaScriptpolyfill的技術完全實作。
* **SSE發送的事件可能包含以下欄位：**
* Event：對應於主題
* Data：主要內容 (文字)
* ID：可用於驗證消息的順序或在發生連接問題時重新發送消息
* retry：用於控制重新連接
* comment：通用字串
* **SSE使用方式區分下列幾種：**
* 所有 Client 都可同步接收到一樣內容 broadcast unnamed event
* 同一群組都可同步接收到一樣內容 broadcast named event
* 單一Client 接收到此內容SseEventSink.send() 或broadcast named event此處 name 此有此 Client 知道。(可能較耗資源，適用於Client少或連線時間較短狀況)
* **SSE安全性相關議題：**
* event.origin
* session (需同一WAR檔)
* named event (use SESSIONID, OTP, UUID as name)
* CORS
* **SSE常搭配 Notification API 一起應用 (範例說明)**