JSON WEB TOKEN

TSAI TUNG-HAN

JSON Web Token簡介

- JSON Web Token (JWT) 是一種開放標準,用於在網路應用程式之間安全地傳輸資訊。
- 它提供了一種簡潔而獨立的方式,使用JSON物件來安全地傳輸資訊。
- 透過JSON Web Token,可以確保資訊完整性和真實性,是目前最流行的API身份驗證解決方案之一。

瀏覽器 BASIC 驗證 — 無法登出

伺服器資料驗證 — Session Base 身份驗證

服務不易擴充，畢竟 Session 資料是跟著服務的，

不同台服務間資料是不互通的。

資料庫驗證 — 史前時代會這樣做

但想像一下，每分鐘有多少人打你驗證身份的 DB，如果你這樣設計服務，對不起~~我們不是朋友

不要驗證 — ???

JWT的出現背景(身份驗證)

JWT的出現背景(服務架構)

微服務

多個微服務中，需要多次的驗證、取得客戶資料

有種你去撈 DB 我保證 DB 負責人會掐死你

RESTful API

RESTful API 是基於無狀態為前提的服務(Stateless Service)

服務的水平擴充

有狀態的服務意味者服務與使用者之間有著「高度的關聯」。

因此，某一時段內同一人只能使用固定的一個服務

跨語言、平台的需求

JAVA – Python

Flask – JAVA

JAVA – node.js

Web Browser 1

Application Server 1

Web Browser 2

Application Server 2

解決方案

真的不要驗證

或著說不要用 DB 驗證

也不要依靠伺服器儲存的資料驗證

用調味料驗證(認真的後面會解釋)

JWT概觀

不在將資料放在伺服器端

- 由使用者負責記錄我是誰，有哪些權限

減少網路傳輸次數

- 省掉了撈資料驗證的過程(DB-data、Session-data)

可信任與安全的

- 透過(加鹽)及不可逆的加密方式，降低被破解的可能性
- 不適合放有風險的資訊 (密碼、信用卡...)

良好的擴充性

- Server 不在需要記錄使用者資訊，因此不會有某人的資料只在某Server 上

  → 對平行擴充、微服務都極為友善

鹽 (密碼學)《wiki》

JWT的組成部分

Header

- 包含基本資訊如類型(typ)和加密演算法(alg)

Payload

- 存放需要傳遞的資料,例如使用者 ID、到期時間等

Signature

- 由 Header、Payload 和密鑰經過**不可逆**的演算法計算獲得
- 用於驗證資料的完整性。

JWT的組成部分(Payload)

Payload

所有不想放 DB 的資料

1. 我是誰 (帳號)
2. 我能幹嘛 (權限)
3. 我還活著嗎 (過期)

然後用 BASE64 加密

PAYLOAD: DATA


```
{
  "Account": "00891341",
  "name": "TSAI TUNG-HAN",
  "role": "PG",
  "exp": "2024-05-24 14:00 "
}
```


BASE64

eyJBY2NvdW50IjoiMDA4OTEzNDEiLCJuYW1lIjoiVFNBSSBUVU5HLUhBTiIsInJvbGUiOiJQRyIsImV4cCI6IjIwMjQtMDUtMjQgMTQ6MDAgIn0

BASE64 加密(無關安全,為防止特殊字元)

JWT的組成部分(Header)

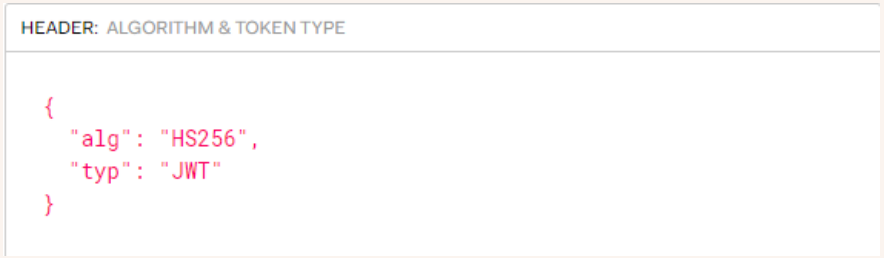
Header

JWT的標頭部分包含

1. 類型資訊(typ)
2. 演算法資訊(alg)。

用於描述JWT的基本特性。

然後用 BASE64 加密



eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

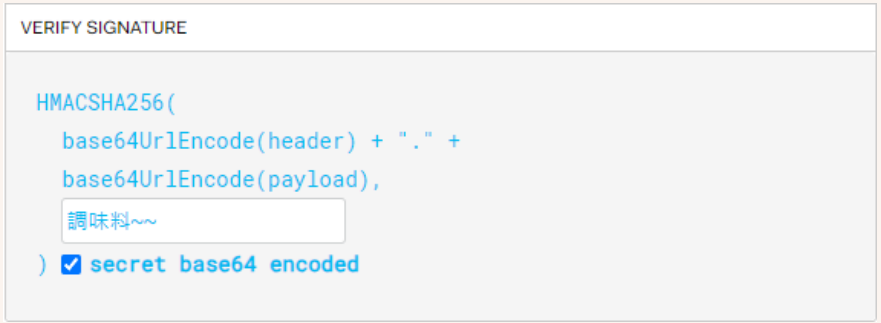
BASE64 加密(無關安全,為防止特殊字元)

JWT的組成部分(Signature)

Signature

[Header + Payload + Key(調味料~~)]經不可逆的演算處理後產生簽名，

未來將用來讓後端驗證資料



SP4EAGR1Q_a-aAgWsHCGB8JQOStVjOPz9AfayipK_Pw

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JBY2NvdW50IjoiMDA4OTEzNDEiLCJuYW1lIjoiV
FNBSSBUVU5HLUhBTiIsInJvbGUiOiJQRyIsImV4
cCI6IjIwMjQtMDUtMjQgMTQ6MDAgIn0.SP4EAGR
1Q_a-aAgWsHCGB8JQOStVjOPz9AfayipK_Pw
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "Account": "00891341",
  "name": "TSAI TUNG-HAN",
  "role": "PG",
  "exp": "2024-05-24 14:00 "
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  調味料~~
) secret base64 encoded
```

XXX.config

試試看吧:《 jwt.io》

Web Browser 1

Application Server 1

Web Browser 2

Application Server 2

生成 JWT 令牌

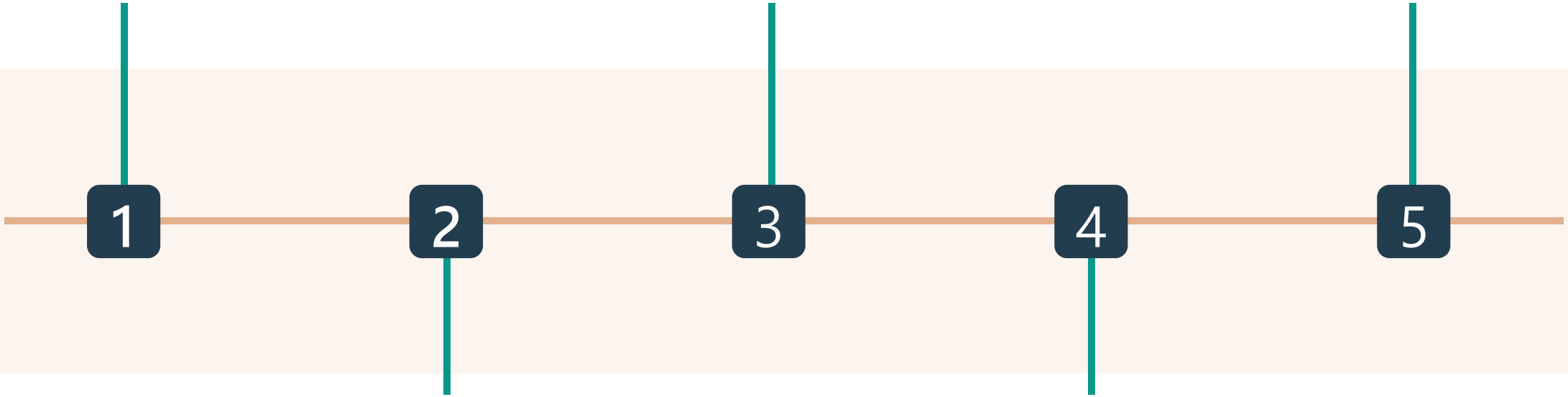
用戶成功登入後, 後端會根據用戶資訊生成一個 JWT 令牌。

附帶 JWT 令牌

當**客戶端**發起請求時,會在請求頭中附帶 JWT 令牌。

根據權限授予存取

通過 JWT 驗證後,根據 JWT 中包含的資訊決定授予的權限。



傳送 JWT 令牌

此 JWT 令牌隨後會被傳回給客戶端, 供後續請求使用。

驗證 JWT 合法性

後端收到請求後會使用預先共享的密鑰驗證 JWT 的合法性。

JWT的優缺點

優點

無狀態、可擴展、跨平台、具有安全性的身份驗證機制。可以輕鬆實現單點登入和權限管理。

缺點

載荷資訊需要進行 Base64 編碼,容易被竊取。過期後無法被撤回,需要引入黑名單來解決。

安全考量

JWT token 一旦被盜用,會造成權限濫用的風險。需要謹慎管理 token 的生命週期和防禦機制。

JWT的安全性考量

竊取 JWT 令牌的風險

一旦 JWT 令牌被惡意獲取,攻擊者可以利用其中包含的權限信息進行未經授權的存取。應用程式必須確保 JWT 的安全傳輸和存儲。

加強傳輸安全性

建議在傳輸 JWT 時使用HTTPS協議,並啟用 SSL/TLS 加密,防止中間人攻擊竊取令牌。此外也需要防範 CSRF 攻擊。

管理 JWT 生命週期

JWT 一旦簽發就很難撤銷,因此需要謹慎控制其生命週期。制定合理的過期時間策略,並採用適當的刷新機制來延長 JWT 的使用期限。

摘要

JWT（JSON Web Tokens）是一種用於網絡應用之間安全地傳遞資訊的開放標準。它主要用於身份驗證和授權，結構包括三部分：頭部（指定類型和加密算法）、負載（包含聲明，例如用戶資訊和權限）和簽名（確保數據未被篡改）。JWT的自包含性質使其在多服務架構中非常有效。

感謝您

TSAI TUNG-HAN