

Coworking App -Backend- JWT Token

Ivan <<ossan>> Pesenti
Software Developer



HACKERSSEN

POWERED BY



TOKEN... TOKEN...
TOKEN OVUNQUE!



Prima dei Token...

- ❑ Prima che fosse introdotta l'autenticazione tramite i token, il meccanismo più usato era quello basato sulle **password**
- ❑ Quest'ultima metodologia implicava i seguenti aspetti:
 - ❑ L'utente doveva generare molte password
 - ❑ L'utente doveva ricordarsi tutte le varie password
 - ❑ L'utente, ogni volta che doveva accedere, doveva immettere le password



HACKERSGEN

POWERED BY
SORINT

Svantaggi delle Password

- ❑ Le persone tendono a scriverle su **file** o **carta**. Questo facilita i furti di credenziali
- ❑ Le persone tendono a ripetere la **stessa** password per più account. In caso di compromissione di un account, a cascata, ne vengono compromessi molti altri
- ❑ Le persone, quando richiesto loro di modificare la password, tendono a **cambiare solamente un carattere**. Questo indebolisce la sicurezza e facilita l'hacker
- ❑ Aumenta il consumo di **memoria** sul server in quanto ogni volta che un utente si logga viene creato un record per la sua transazione



HACKERSGEN

POWERED BY

SORINT

AUTENTICAZIONE BASATA SUI TOKEN



Autenticazione a Token

- ❑ Funzionano come i **biglietti** per i mezzi pubblici
- ❑ Trascorso il periodo di validità (può dipendere dal numero di accessi e/o dal periodo di tempo), il token deve essere **rigenerato**
- ❑ Deve essere incluso in ogni richiesta che il client invia al server
- ❑ Solitamente, viene passato in un header della richiesta HTTP
- ❑ È una semplice stringa di testo che contiene delle informazioni sull'utente
- ❑ È il meccanismo più usato per l'autenticazione e/o autorizzazione



HACKERSGEN

POWERED BY

SORINT

Il Flusso



okta



HACKERSGEN
POWERED BY
SORINT

TOKEN JWT



HACKERSGEN

POWERED BY
SORINT

Cosa Sono?

- ❑ **JSON Web Token** (o **JWT**) è uno standard libero
- ❑ È una stringa compatta che viene passata da un sistema all'altro durante la comunicazione
- ❑ Contiene informazioni riguardanti l'utente e i suoi permessi. Le informazioni scritte sul token vengono chiamate **claims**
- ❑ È composto da tre parti:
 - ❑ **Header**: definisce il tipo di token e l'algoritmo per la sua firma digitale
 - ❑ **Payload**: contiene l'emittente del token, la data di scadenza e altre informazioni
 - ❑ **Signature**: verifica che il messaggio non è stato cambiato durante il transito da un punto all'altro




HACKERSGEN

POWERED BY
SORINT



JSON Web Tokens - jwt.io

jwt.io

Ryan

 **JWT**

Debugger Libraries Ask Get a T-shirt!

ALGORITHM

HS256

Encoded

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0Ij0iMTYzNDU2Nzg5LjVJA950rM7E2cBab30RMhRHdCEfxjoYZgeFONFh7HgQ
```

Decoded

HEADER:

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

PAYLOAD:

```
{  "sub": "1234567890",  "name": "John Doe",  "admin": true}
```

VERIFY SIGNATURE

HMACSHA256(
base64UrlEncode(header) + "." +
base64UrlEncode(payload),

secret

) ☐ secret base64 encoded

Signature Verified



HACKERSGEN

POWERED BY
SORINT

TOKEN IN GO



HACKERSGEN

POWERED BY
SORINT

THANKS!

@ossan



HACKERSGEN

POWERED BY

