JSON Web Tokens

Brežnjak Nikola
Senior Software Engineer
TelTech Systems, Inc

What is JWT

Just White T-shirt

JWT = JSON Web Token (jot)

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiYTFiMmMzIiwidXNlcm5hbWUiOiJuaWtvbGEifQ==.mKIuU0V0Bo99JU5XbeMe6g-Hrd3ZxJRlmdHFrEkz0Wk

- header - information about the token
- payload - data transmitted between two parties
- verification signature - verify that data has not changed

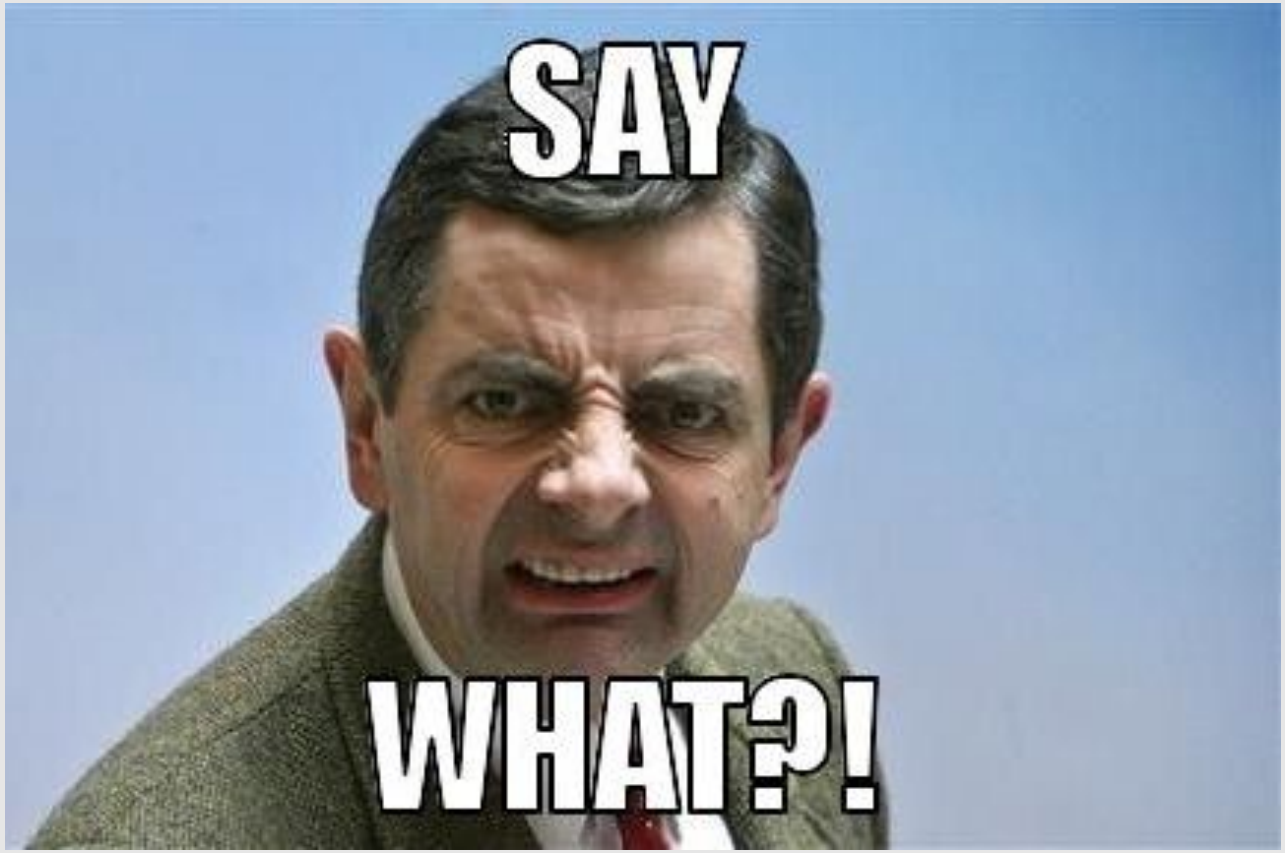
What do the officials say

Same ol same ol

JSON Web Tokens are an open, industry standard RFC 7519 method for representing claims securely between two parties.

Huh!?

I always loved theory



Let’s start again

Second time is a charm :)

- JWTs are used to securely communicate information between two parties

- A **claim** is some data that is sent along with the token (`user_id`)

Secure communication

NSA is watching

- Secure communication === information has not been tampered
- Secure communication !== hidden from a potential attacker

Authentication !?

To be or not to be

- Useful for authentication
- Short and easy to send via POST, HTTP header, or added as a query string

- ⚠️ NoToJWT_1 & NoToJWT_2

Learn by doing

~ be 1% better than yesterday

Payload

```
{
    "user_id": "a1b2c3",
    "username": "nikola"
}
```

JavaScript

every ad is a JavaScript ad

```
btoa(JSON.stringify({
    "user_id": "a1b2c3",
    "username": "nikola"
}));
```

Go

The dream came true

```go
package main

import (
    "encoding/base64"
    "fmt"
)

func main() {
    data := `{"user_id":"a1b2c3","username":"nikola"}`
    uEnc := base64.URLEncoding.EncodeToString([]byte(data))
    fmt.Println(uEnc)
}
```

Encoded payload

One down, two to go!

eyJ1c2VyX2lkIjoiYTFiMmMzIiwidXNlcm5hbWUiOiJuaWtvbGEifQ==

Decoded payload

What goes up must come down

JavaScript

```
atob('eyJ1c2VyX2lkIjoiYTFiMmMzIiwidXNlcm5hbWUiOiJuaWtvbGEif
Q==')
```

Go

```
uDec, _ :=
base64.URLEncoding.DecodeString("eyJ1c2VyX2lkIjoiYTFiMmMzIi
widXNlcm5hbWUiOiJuaWtvbGEifQ==")
```

Output

```
{"user_id":"a1b2c3","username":"nikola"}
```

Encoded header

Think with your head

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

// Base64 encoded header
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
```

[Tutorial about different hashing algorithms in JWT]

Verification signature

Think with your head

```
HS256([header].[payload], secret)

[header].[payload]=
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiYTFiMmMzIiwi
dXNlcm5hbWUiOiJuaWtvbGEifQ==

secret = 42isTheAnswer
```

JavaScript

every ad is a JavaScript ad

⚠️ npm install base64url

```
var base64url = require('base64url');

var crypto    = require('crypto');
var message     = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiYTFiMmMzIiwidXNlcm5hbWUiOiJuaWtvbGEifQ==';
var key       = '42isTheAnswer';
var algorithm = 'sha256';
var hash, hmac;

hmac = crypto.createHmac(algorithm, key);
hmac.setEncoding('base64');
hmac.write(message);
hmac.end();
hash = hmac.read();

var final = base64url.fromBase64(hash);
console.log(final)
```

Go

The dream came true ~every backend dev

```
package main

import (
    "crypto/hmac"
    "crypto/sha256"
    "encoding/base64"
    "fmt"
)

func main() {
    message := "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiYTFiMmMzIiwidXNlcm5hbWUiOiJuaWtvbGEifQ=="
    sKey := "42isTheAnswer"

    key := []byte(sKey)
    h := hmac.New(sha256.New, key)
    h.Write([]byte(message))
    b := base64.URLEncoding.EncodeToString(h.Sum(nil))
    fmt.Println(string(a))
}
```

We're done!

3/3, A+

Verification signature

mKIuU0V0Bo99JU5XbeMe6g-Hrd3ZxJRlmdHFrEkz0Wk

Final JWT

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiYTFiMmMzIiwi
dXNlcm5hbWUiOiJuaWtvbGEifQ==.mKIuU0V0Bo99JU5XbeMe6g-
Hrd3ZxJRlmdHFrEkz0Wk

Test it on https://jwt.io/

How is this secure!?

Are you trying to trick me kid!?

- Payload and Header are encoded, not encrypted
- It can’t be changed

JWTs in authentication

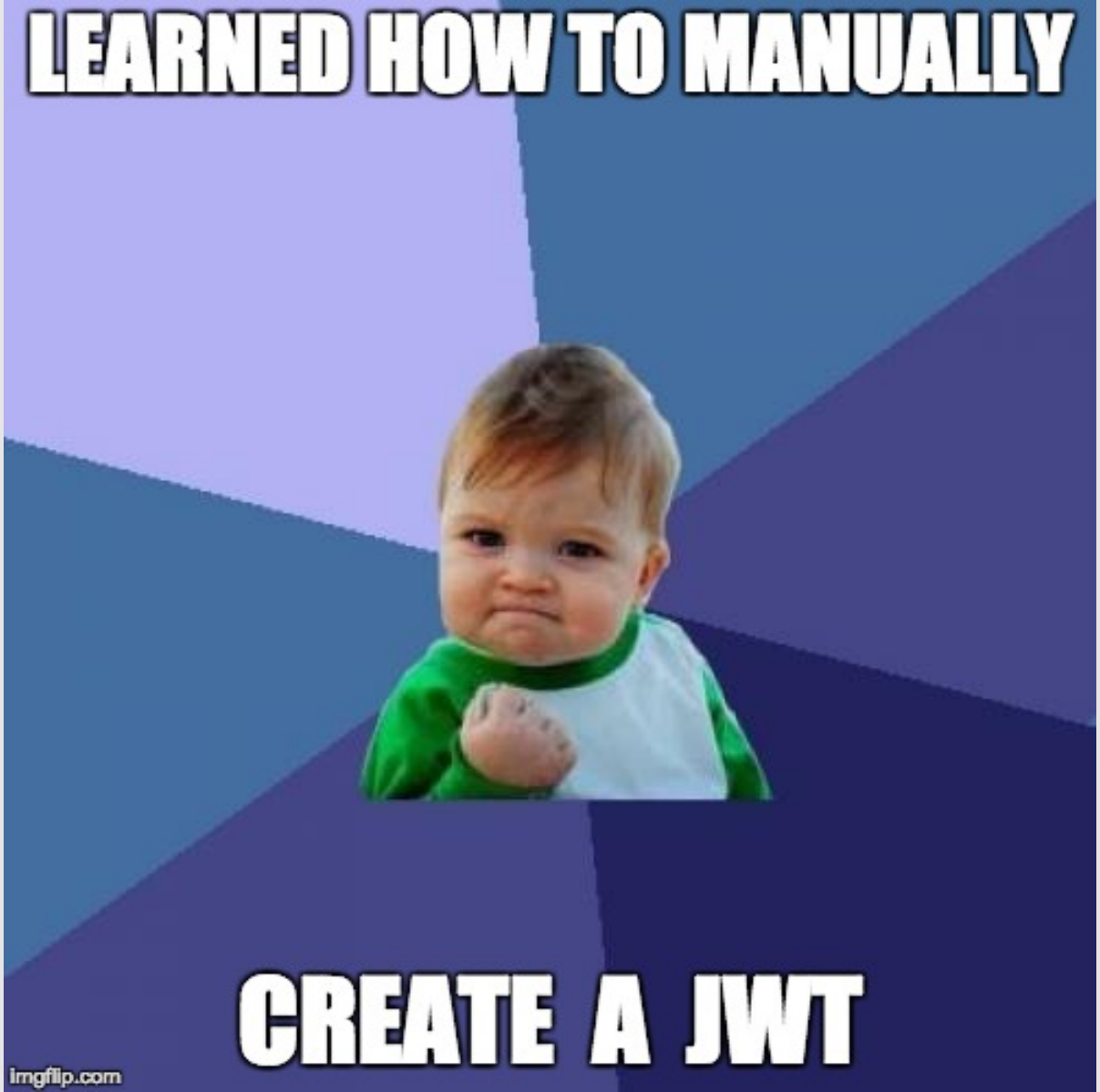
Are you trying to trick me kid!?

- user ID and an expiration timestamp == Bearer Token
- A bearer token is a signed temporary replacement for the username/password combination
- Login page purpose is to give the user this token
- Key property of JWTs is that in order to confirm if they are valid we only need to look at the token itself

What awaits us in the next episode

Milk and cookies for everyone!

JWT authentication in an Angular application with a Go backend


LEARNED HOW TO MANUALLY
CREATE A JWT
imgflip.com