WTH is a JWT

A Quick Intro To Token Based Authentication

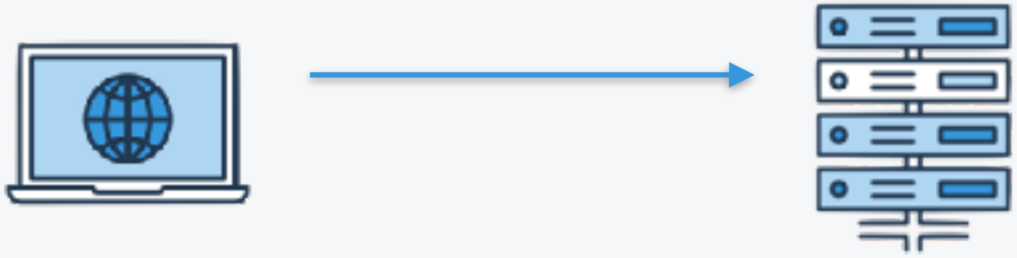
About Me

@joel__lord

joellord

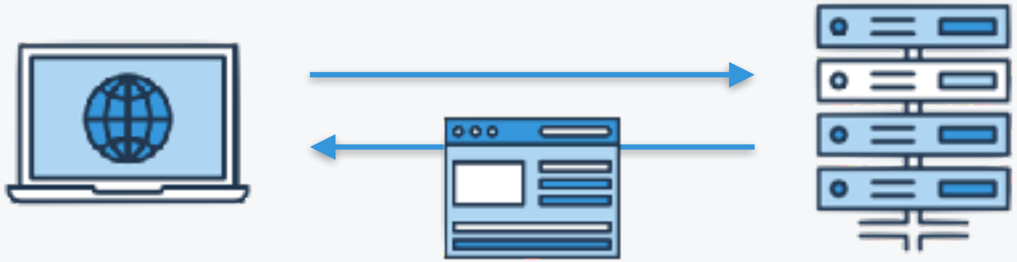
Traditional Applications

- Browser requests a login page



Traditional Applications

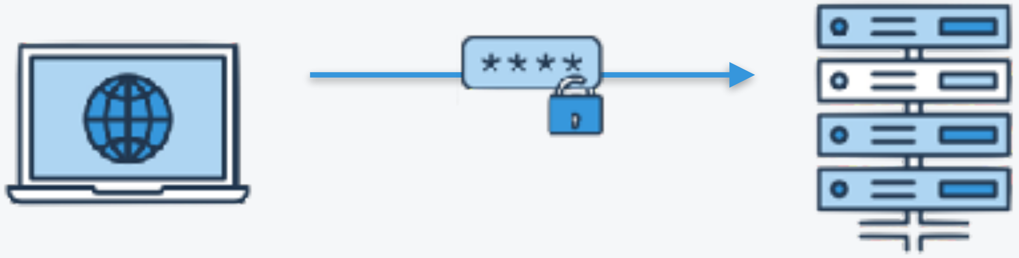
- Browser requests a login page



Auth0

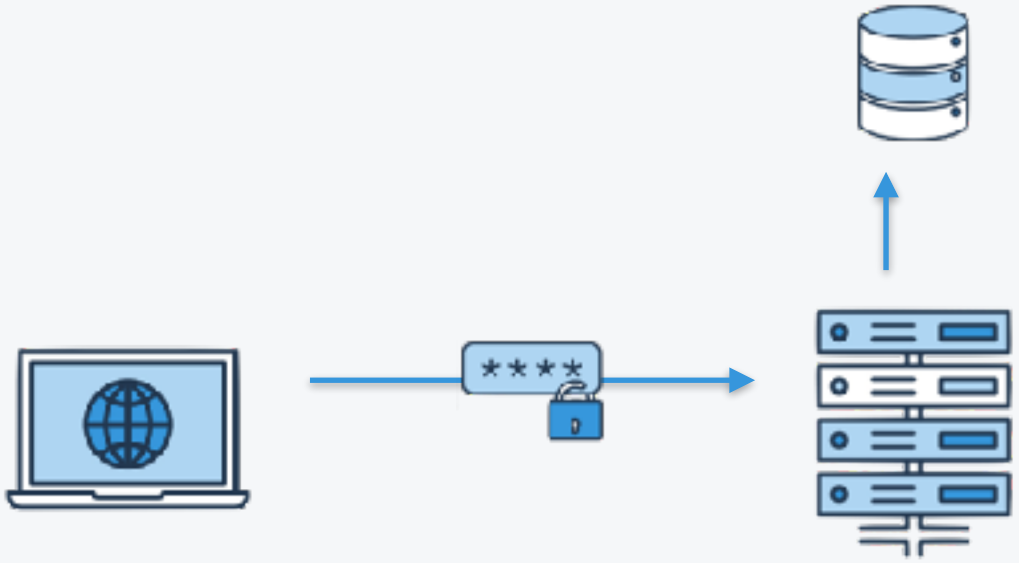
Traditional Applications

- Browser requests a login page



Traditional Applications

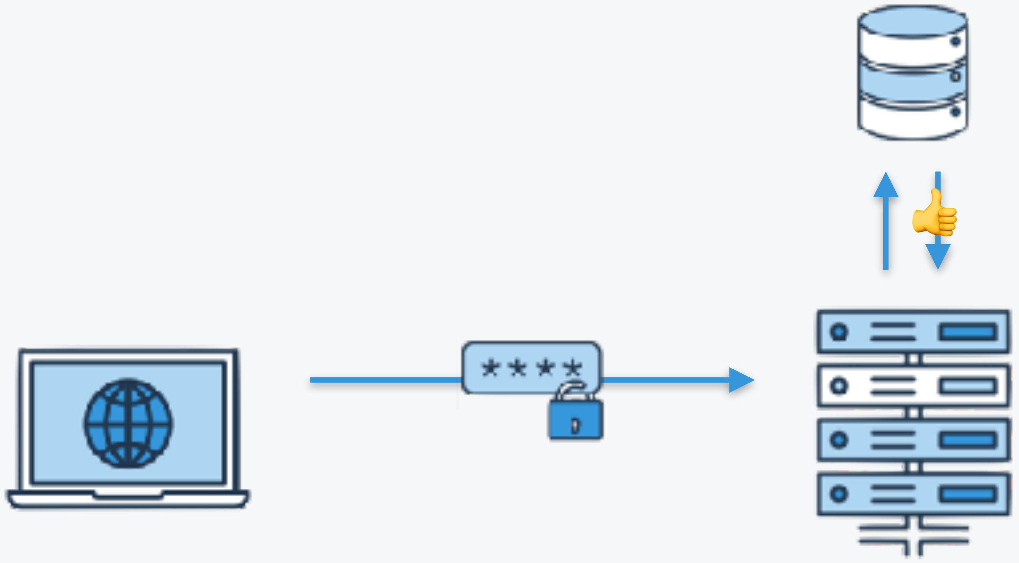
- Browser requests a login page
- The server validates on its database



Auth0

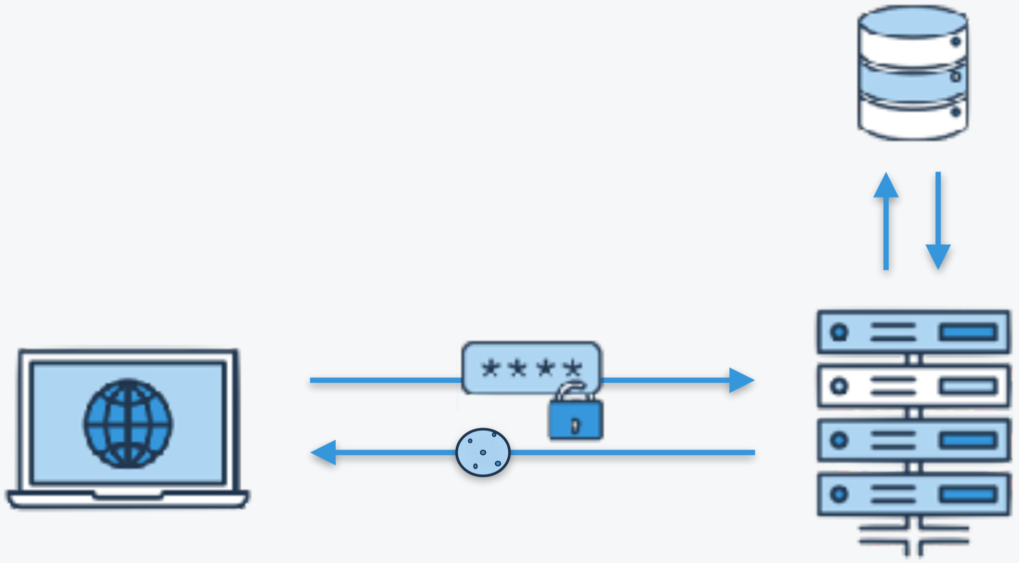
Traditional Applications

- Browser requests a login page
- The server validates on its database



Traditional Applications

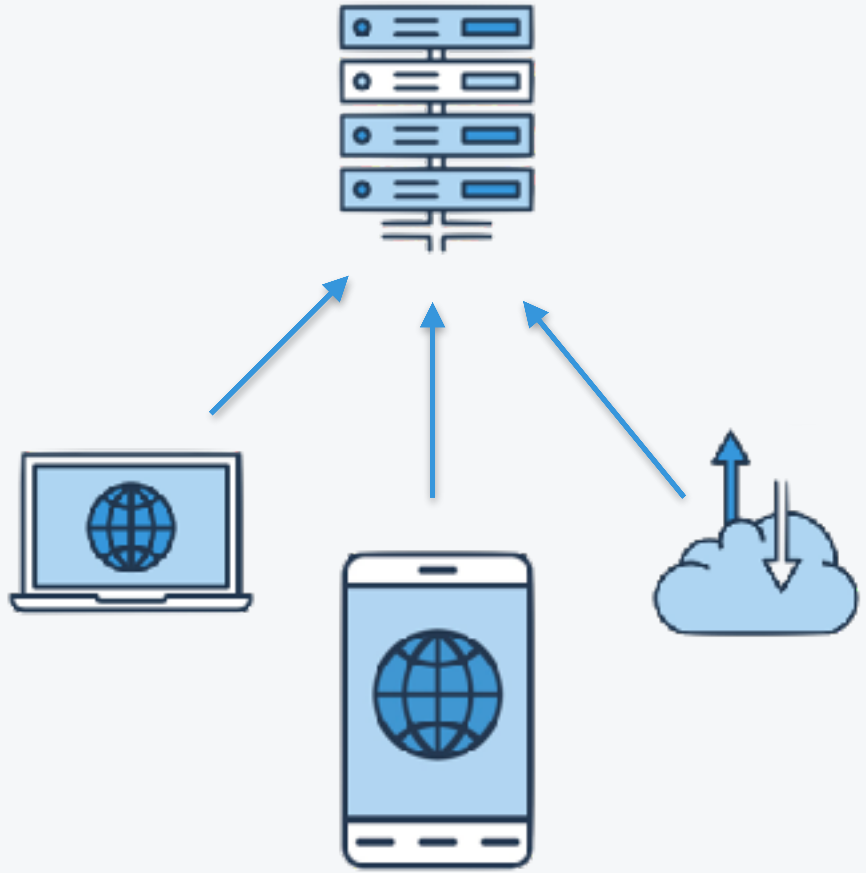
- Browser requests a login page
- The server validates on its database
- It creates a session and provides a cookie identifier



Auth0

What’s wrong with traditional auth?

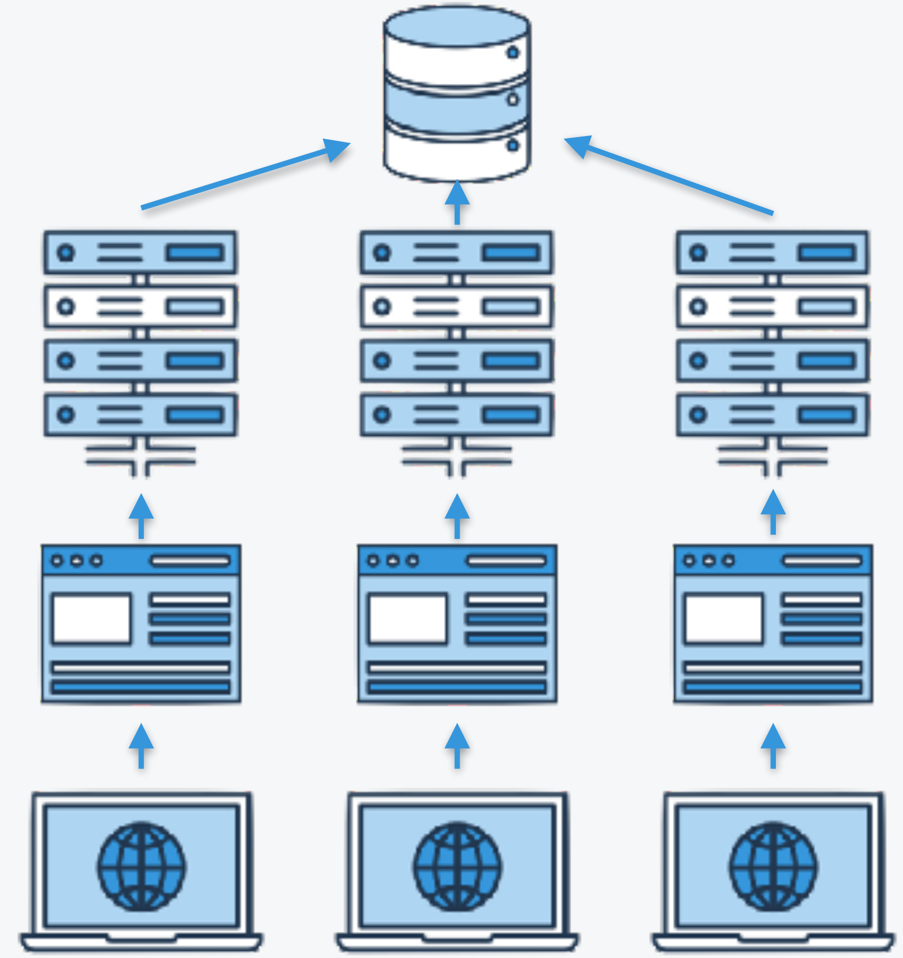
- Multiple platforms connecting to your application



Auth0

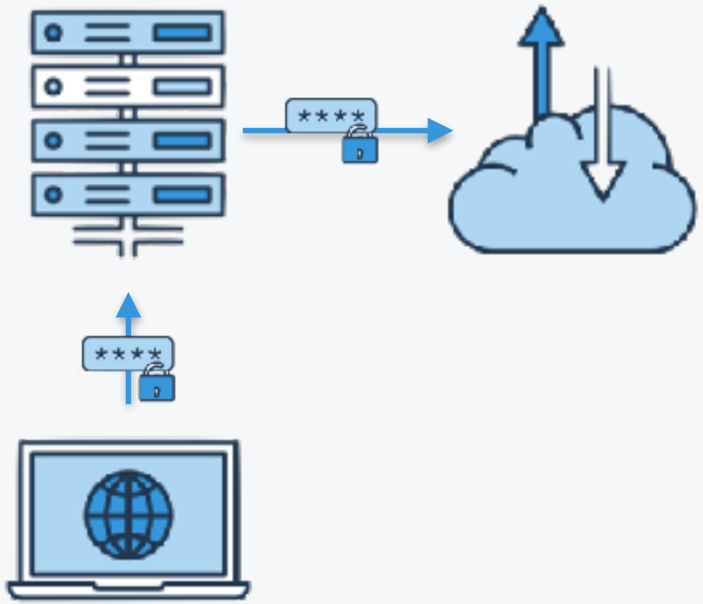
What’s wrong with traditional auth?

- Multiple platforms connecting to your application
- Tightly coupled



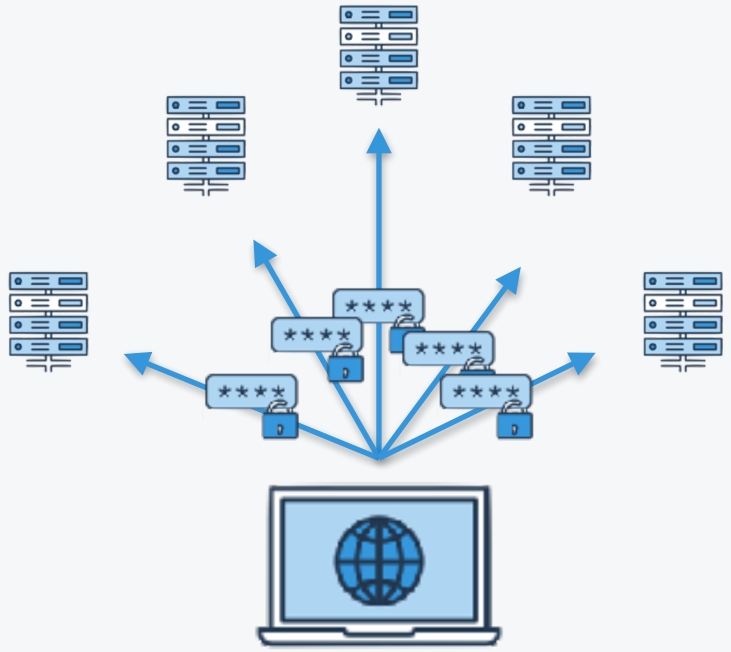
What’s wrong with traditional auth?

- Multiple platforms connecting to your application
- Tightly coupled
- Sharing credentials to connect to another API



What’s wrong with traditional auth?

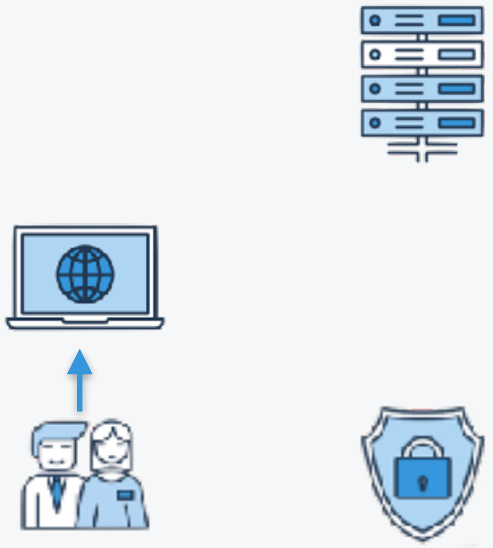
- Multiple platforms connecting to your application
- Tightly coupled
- Sharing credentials to connect to another API
- Users have a gazillion passwords to remember, which increases security risks



OAuth

Authentication Flows

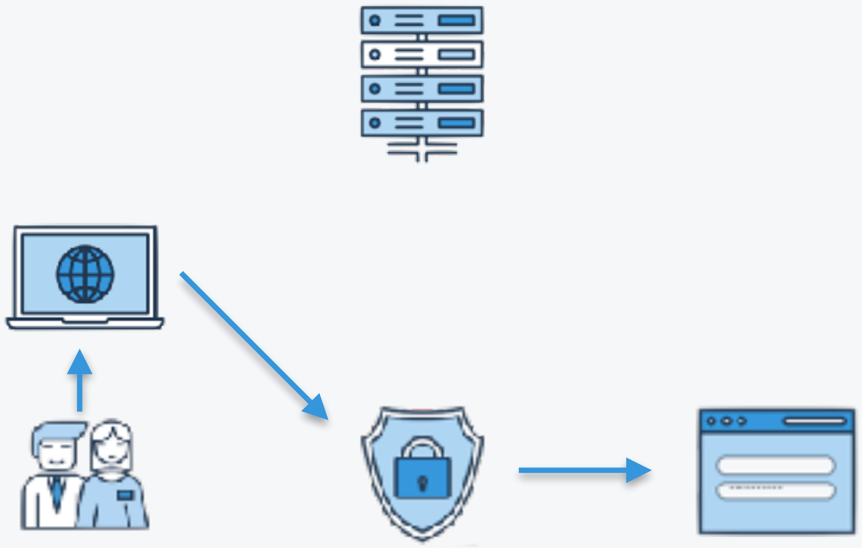
Implicit Flow



@joel__lord
#OSNorth

Authentication Flows

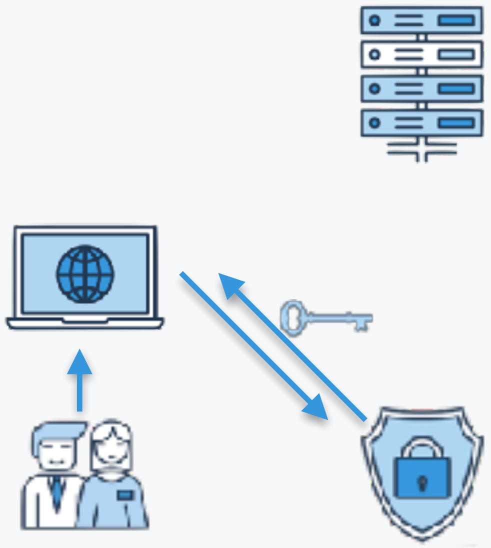
Implicit Flow



@joel__lord
#OSNorth

Authentication Flows

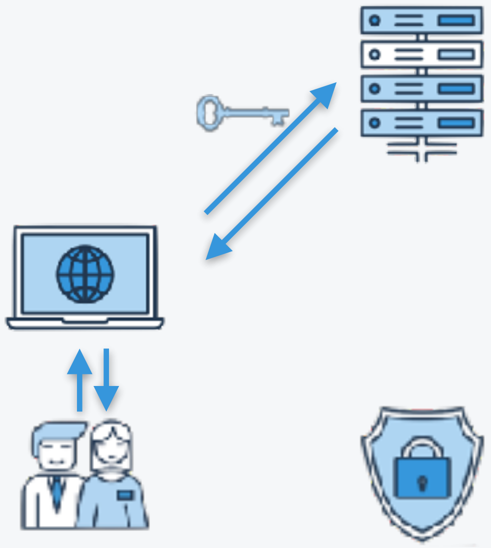
Implicit Flow



@joel__lord
#OSNorth

Authentication Flows

Implicit Flow



@joel__lord
#OSNorth

What is a Token?

JSON Web Token

- Header
- Payload
- Signature

Header

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

Payload

```
{
  "sub": "1234567890",
  "name": "Joel Lord",
  "scope": "posts:read posts:write"
}
```

Signature

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

JSON Web Token

- Header
- Payload
- Signature

Header

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

Payload

eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvZWwgTG
9yZCIsImFkbWluIjp0cnVlLCJzY29wZSI6InBvc3RzOnJlY
WQgcG9zdHM6d3JpdGUifQ

Signature

XesR-pKdlscHfUwoKvHnACqfpe2ywJ6t1BJKsq9rEcg

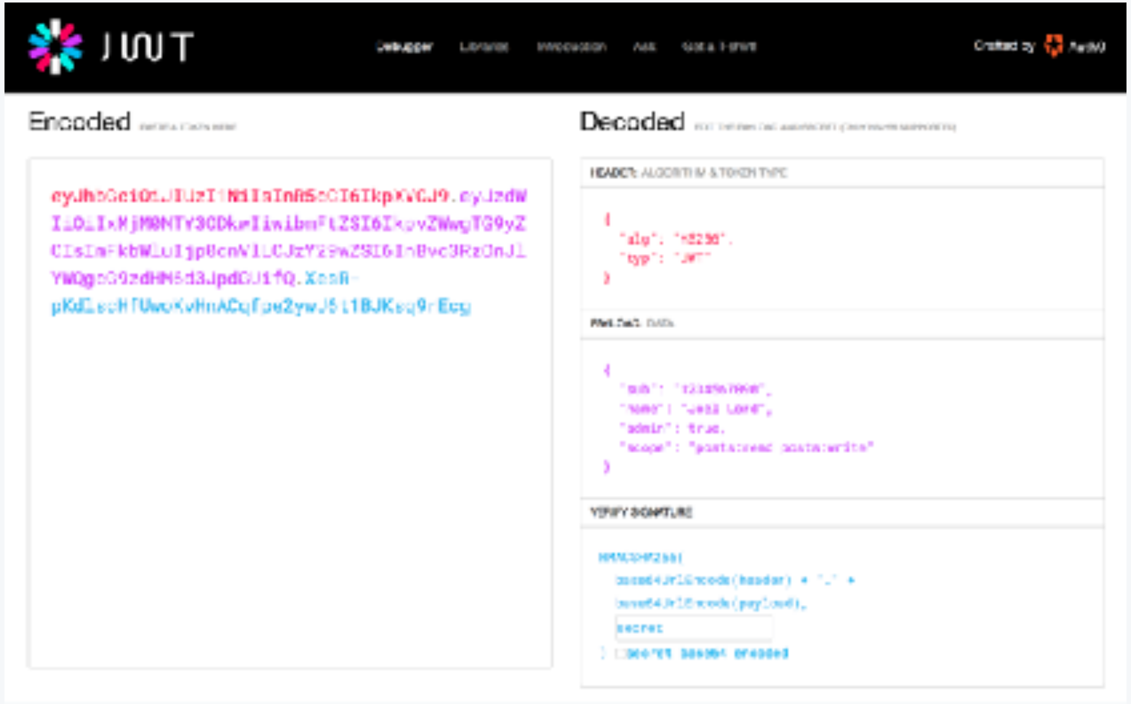
JSON Web Token

- Header
- Payload
- Signature

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvZWwgTG9yZCIsImFkbWluIjp0cnVlLCJzY29wZSI6InBvc3RzOnJlYWQgcG9zdHM6d3JpdGUifQ.XesR-pKdlscHfUwoKvHnACqfpe2ywJ6t1BJKsq9rEcg
```

JSON Web Token

- Header
- Payload
- Signature



Encoded

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdW
IiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvZWwgTG9yZ
CIsImFkbWluIjp0cnVlLCJzY29wZSI6InBvc3RzOnJl
YWQgcG9zdHM6d3JpdGUifQ.XsnR-
pKdLscHTUwoKvHnACqfpe2ywJ5i1BJKsq9rEcg
```

Decoded

HEADER: ALGORITHM & TOKEN TYPE

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

PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "Joel Lord",
  "admin": true,
  "scope": "posts:read posts:write"
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret
) ☐ secret base64 encoded
```

Image: https://jwt.io

Auth0

Code

Authentication

Create a JWT

Create a JWT

```
// sign with default (HMAC SHA256)
var jwt = require('jsonwebtoken');
var token = jwt.sign({ name: 'Joel Lord' }, 'secret');
```

Create a JWT

```
// sign with default (HMAC SHA256)
var jwt = require('jsonwebtoken');
var token = jwt.sign({ name: 'Joel Lord' }, 'secret');
```

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJuYW1lIjoiSm9lbCBMb3JkIiwiaWF0Ijox
NTI1MTc2NDI3fQ.V89hohVfp1uVNfunkpdl
ewNyvGCX5iPPxe1YpM-RqRg
```

Create a JWT

```
// sign with default (HMAC SHA256)
var jwt = require('jsonwebtoken');
var token = jwt.sign({ name: 'Joel Lord' }, 'secret');
```

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoiSm9lbCBMb3JkIiwiaWF0Ijox
NTI1MTc2NDI3fQ.V89hohVfp1uVNfunkpdl
ewNyvGCX5iPPxe1YpM-RqRg

```
{
  "name": "Joel Lord",
  "iat": 1525176427
}
```

API

Validate a JWT

Validate a JWT

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoiSm9lbCBMb3JkIiwiaWF0IjoxNTI1MTc2NDI3fQ
.V89hohVfp1uVNfunkpdlewNyvGCX5iPPxe1YpM-RqRg
```

Validate a JWT

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoiSm9lbCBMb3JkIiwiaWF0IjoxNTI1MTc2NDI3fQ
.V89hohVfp1uVNfunkpdlewNyvGCX5iPPxe1YpM-RqRg

```
var jwt = require('jsonwebtoken');

// verify a token
var data = jwt.verify(token, 'secret');
console.log(data);
```

Validate a JWT

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoiSm9lbCBMb3JkIiwiaWF0IjoxNTI1MTc2NDI3fQ
.V89hohVfp1uVNfunkpdlewNyvGCX5iPPxe1YpM-RqRg
```

```javascript
var jwt = require('jsonwebtoken');

// verify a token
var data = jwt.verify(token, 'secret');
console.log(data);
```

```
{
  "name": "Joel Lord",
  "iat": 1525176427
}
```

Auth0

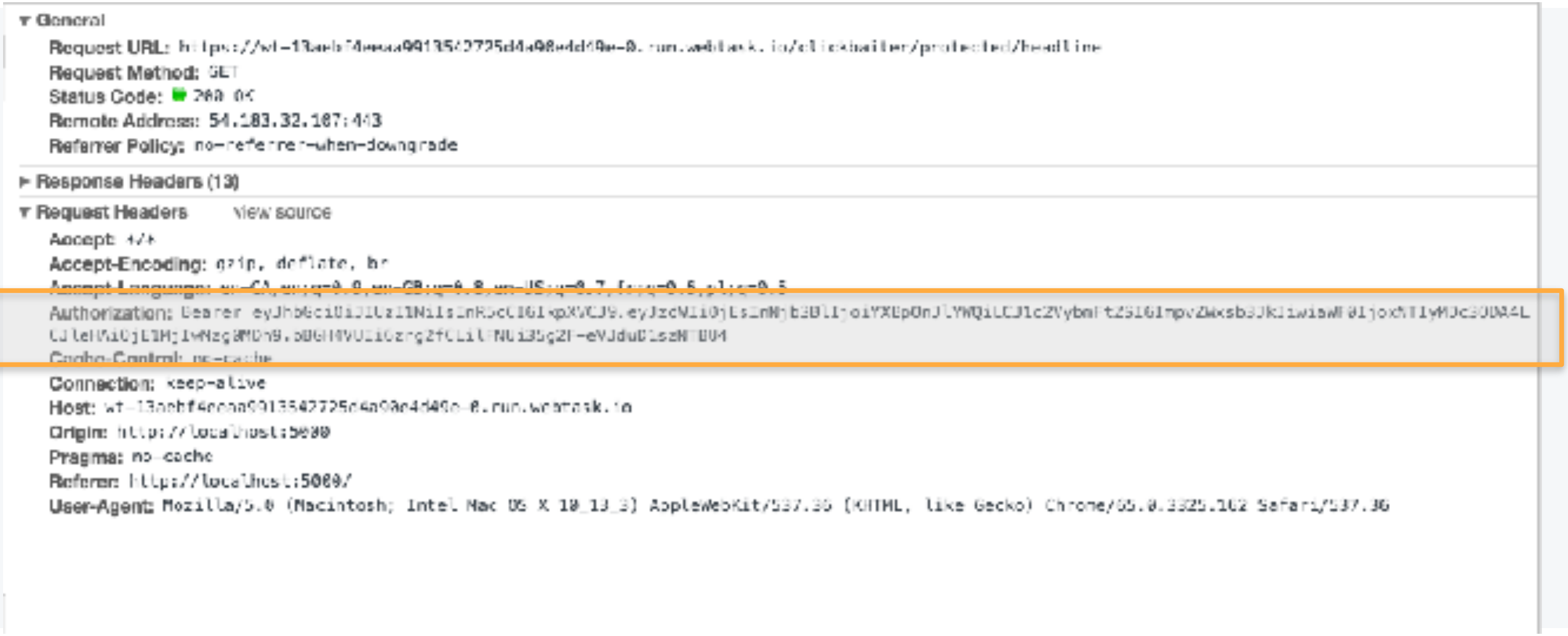
@joel__lord
#OSNorth

Front End

Handle a JWT

Front-End

Add the headers



```
▼ General
    Request URL: https://wt-13aeb14eeaa99135427725d4a98edd49e-0.run.webtask.io/clickbaiter/protected/headline
    Request Method: GET
    Status Code: 🟩 200 OK
    Remote Address: 54.183.32.187:443
    Referrer Policy: no-referrer-when-downgrade
▶ Response Headers (13)
▼ Request Headers        view source
    Accept: */*
    Accept-Encoding: gzip, deflate, br
    Accept-Language: en-CA,en;q=0.9,en-GB;q=0.8,en-US;q=0.7,fr;q=0.6,pl;q=0.5
    Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOjEsInNjb3BlIjoiYXBpOnJlYWQiLCJ1c2VybmFtZSI6ImpvZWxsb3JkIiwiaWF0IjoxNTIyMDc3ODA4L
    CJleHAiOjE1MjIwNzg4MDh9.5BGHMVUIi0zrg2fCLiLFNUi35g2F-eVJduBIszNTBU4
    Cache-Control: no-cache
    Connection: keep-alive
    Host: wt-13aebf4eeaa99135427725d4a98e4d49e-0.run.webtask.io
    Origin: http://localhost:5000
    Pragma: no-cache
    Referer: http://localhost:5000/
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.162 Safari/537.36
```

@joel__lord
#OSNorth

Live Demo

https://github.com/joellord/secure-spa-auth0

Delegation!

WTH is a JWT

Open Source North
Minneapolis, MN
June 14th, 2018

@joel__lord

joellord