

API Authorization Fundamentals



Cory House

PRINCIPAL CONSULTANT

@housecor reactjsconsulting.com



Agenda



Create APIs via Node and Express

Integrate APIs with create-react-app

Configure Express to parse JWTs

Create multiple API endpoints

- Public (anyone can call)
- Private (login required)



Validating a JWT: 2 Steps



Verify Signature



Validate Claims

Step 1: Verify JWT Signature

```
{
  "keys": [
    {
      "alg": "RS256",
      "kty": "RSA",
      "use": "sig",
      "x5c": [
        "MIIC..."
      ],
      "n": "ydmJ...",
      "e": "AQAB",
      "kid": "NUYzM0ZCMTFFQkMwNzVDM0Y0QTFGNjAyMkU2NDBBRDdGNDY4RTc2NA",
      "x5t": "NUYzM0ZCMTFFQkMwNzVDM0Y0QTFGNjAyMkU2NDBBRDdGNDY4RTc2NA"
    }
  ]
}
```

https://YOUR_AUTH0_DOMAIN/.well-known/jwks.json

JSON Web Key Set (JWKS)

[In this article](#) 

When creating applications and resources servers (APIs) in Auth0, two algorithms are supported for signing [JSON Web Tokens \(JWTs\)](#): RS256 and HS256. RS256 generates an asymmetric signature, which means a private key must be used to sign the JWT and a different public key must be used to verify the signature.

Auth0 uses the [JWK](#) specification to represent the cryptographic keys used for signing RS256 tokens. This specification defines two high level data structures: **JSON Web Key (JWK)** and **JSON Web Key Set (JWKS)**. Here are the definitions directly from the specification:

Item	Description
JSON Web Key (JWK)	A JSON object that represents a cryptographic key. The members of the object represent properties of the key, including its value.
JSON Web Key Set (JWKS)	A JSON object that represents a set of JWKs. The JSON object MUST have a <code>keys</code> member, which is an array of JWKs.

At the most basic level, the JWKS is a set of keys containing the public keys that should be used to verify any JWT issued by the authorization server. Auth0 exposes a JWKS endpoint for each tenant, which is found at `https://YOUR_AUTH0_DOMAIN/.well-known/jwks.json`. This endpoint will

Step 2: Validate the Claims

exp
iss
aud

Expiration
Issued by
Audience

Confirm it hasn't expired
Confirm it matches your Auth0 domain
Confirm it matches your clientID



Authenticating Via JWT

Authorization: Bearer <token>



Summary



Created APIs via Node and Express

Configured Express to parse JWTs

Created API endpoints

- Public (anyone can call)
- Private (login required)

Next up:

API Authorization with scopes and rules

