+++ title = “JWT Authentication” description = “Grafana JWT Authentication” keywords = [“grafana”, “configuration”, “documentation”, “jwt”, “jwk”] weight = 250 +++

JWT authentication

You can configure Grafana to accept a JWT token provided in the HTTP header. The token is verified using any of the following:

- PEM-encoded key file
- JSON Web Key Set (JWKS) in a local file
- JWKS provided by the configured JWKS endpoint

Enable JWT

To use JWT authentication:

1. Enable JWT in the [main config file]({{< relref “../administration/configuration.md” >}}).
2. Specify the header name that contains a token.

```
[auth.jwt]
# By default, auth.jwt is disabled.
enabled = true

# HTTP header to look into to get a JWT token.
header_name = X-JWT-Assertion
```

Configure login claim

To identify the user, some of the claims needs to be selected as a login info. You could specify a claim that contains either a username or an email of the Grafana user.

Typically, the subject claim called `"sub"` would be used as a login but it might also be set to some application specific claim.

```
# [auth.jwt]
# ...

# Specify a claim to use as a username to sign in.
username_claim = sub

# Specify a claim to use as an email to sign in.
email_claim = sub

# auto-create users if they are not already matched
# auto_sign_up = true
```

If `auto_sign_up` is enabled, then the `sub` claim is used as the “external Auth ID”. The `name` claim is used as the user’s full name if it is present.

Signature verification

JSON web token integrity needs to be verified so cryptographic signature is used for this purpose. So we expect that every token must be signed with some known cryptographic key.

You have a variety of options on how to specify where the keys are located.

Verify token using a JSON Web Key Set loaded from https endpoint

For more information on JWKS endpoints, refer to Auth0 docs.

```
# [auth.jwt]
# ...

jwk_set_url = https://your-auth-provider.example.com/.well-known/jwks.json

# Cache TTL for data loaded from http endpoint.
cache_ttl = 60m
```

Verify token using a JSON Web Key Set loaded from JSON file

Key set in the same format as in JWKS endpoint but located on disk.

```
jwk_set_file = /path/to/jwks.json
```

Verify token using a single key loaded from PEM-encoded file

PEM-encoded key file in PKIX, PKCS #1, PKCS #8 or SEC 1 format.

```
key_file = /path/to/key.pem
```

Validate claims

By default, only `"exp"`, `"nbf"` and `"iat"` claims are validated.

You might also want to validate that other claims are really what you expect them to be.

```
# This can be seen as a required "subset" of a JWT Claims Set.
expect_claims = {"iss": "https://your-token-issuer", "your-custom-claim": "foo"}
```