

go-oidfed



Go Implementation of OpenID Federation

- <https://github.com/go-oidfed>
- <https://github.com/go-oidfed/lib>
- <https://github.com/go-oidfed/lighthouse>
- <https://github.com/go-oidfed/whoami-rp>
- <https://github.com/go-oidfed/offa>



go-oidfed/lib

- <https://github.com/go-oidfed/lib>
- Go library to enable other applications to add oidfed support.
- Core of other repositories.

go-oidfed/lighthouse

- <https://github.com/go-oidfed/lighthouse>
- Formerly known as myoidc/gota or example TA
- Flexible, configurable Federation Entity
 - Trust Anchor / Intermediate Authority
 - Resolver
 - Trust Mark Issuer
 - Entity Collection Endpoint
 - ...
- Docker Image: oidfed/lighthouse



LightHouse - Configuration

- Detailed explanation of configuration options:
- <https://go-oidfed.github.io/lighthouse/config>
- Configuration was improved and extended
 - Continue to extended configuration options

LightHouse - Configuration - Example [Parts]

```
endpoints:
  fetch:
    path: "/fetch"
  list:
    path: "/list"
  resolve:
    url: "https://external.location.com/resolve"
  trust_mark:
    path: "/trustmark"
    trust_mark_specs:
      - trust_mark_type: "https://tm.example.org"
        lifetime: 3600
        ref: "https://tm.example.org/ref"
        logo_uri: "https://tm.example.org/logo"
        extra_claim: "example"
        delegation_jwt:
      - trust_mark_type: "https://edugain.org"
        lifetime: 86400
  trust_mark_status:
    path: "/trustmark/status"
  trust_mark_list:
    path: "/trustmark/list"
  enroll:
    path: "/enroll"
    checker:
      type: multiple_or
      config:
        - type: trust_mark
          config:
            trust_mark_type: https://tm.example.org
            trust_anchors:
              - entity_id: https://ta.example.org
        - type: trust_mark
          config:
            trust_mark_type: https://tm.example.com
            trust_anchors:
              - entity_id: https://example.com
              - entity_id: https://foo.bar.com
```

LightHouse - Docker Deployment

```
└─ caddy/
  └─ Caddyfile
  └─ config/
  └─ data/
└─ docker-compose.yaml
└─ lighthouse/
  └─ config.yaml
  └─ data/
    └─ metadata-policy.json
    └─ storage/
    └─ signing/
```

docker-compose.yaml

```
services:
  caddy:
    image: caddy:latest
    restart: unless-stopped
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./caddy/Caddyfile:/etc/caddy/Caddyfile
      - ./caddy/data:/data
      - ./caddy/config:/config

  lighthouse:
    image: oidfed/lighthouse:main
    restart: unless-stopped
    volumes:
      - ./lighthouse/config.yaml:/config.yaml:ro
      - ./lighthouse/data:/data
```

caddy/Caddyfile

```
lighthouse.example.com {
  reverse_proxy lighthouse:7672
}
```

lighthouse/config.yaml

```
server:
  port: 7672
signing:
  alg: ES256
  key_file: "/data/signing/signing.key"
federation_data:
  entity_id: "https://lighthouse.example.com"
  federation_entity_metadata:
    display_name: "Example Federation TA"
    organization_name: "Example Organization"
    metadata_policy_file: "/data/metadata-policy.json"
storage:
  backend: badger
  data_dir: "/data/storage"
endpoints:
  fetch:
    path: "/fetch"
  list:
    path: "/list"
  resolve:
    path: "/resolve"
  trust_mark:
    path: "/trustmark"
  trust_mark_specs:
    - trust_mark_type: "https://tm.example.org"
      lifetime: 3600
      ref: "https://tm.example.org/ref"
      logo_uri: "https://tm.example.org/logo"
      checker:
        type: trust_path
        config:
          trust_anchors:
            - entity_id: "https://lighthouse.example.cc"
trust_mark_list:
  path: "/trustmark/list"
```

go-oidfed/offa

Openid
Federation
Forward
Auth



- <https://github.com/go-oidfed/offa>
 - Extensive Documentation: <https://go-oidfed.github.io/offa/>
 - Docker Image: oidfed/offa
-
- Enabling OpenID Federation SSO for “legacy” Services.
 - Forward Authentication Service:
 - OFFA acts as a gatekeeper in front of services, handling authentication requests via a reverse proxy
 - Works with [NGINX](#), [Traefik](#), [Caddy](#)
 - Can also be used with the AuthMemCookie [Apache](#) Module
 - Pass Userinfo to Service via HTTP Headers
 - Easy to deploy with docker compose

OFFA - Openid Federation Forward Auth

docker-compose.yml

```
services:
  caddy:
    image: caddy:latest
    restart: unless-stopped
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./caddy/Caddyfile:/etc/caddy/Caddyfile
      - ./caddy/data:/data
      - ./caddy/config:/config
  offa:
    image: oidfed/offa:main
    restart: unless-stopped
    volumes:
      - ./offa/config.yaml:/config.yaml:ro
      - ./offa:/data
  # This would be your service
  whoami:
    image: containous/whoami
    restart: unless-stopped
```

caddy/Caddyfile

```
offa.example.com {
  reverse_proxy offa:15661
}

whoami.example.com {
  forward_auth offa:15661 {
    uri /auth
    copy_headers X-Forwarded-User \
                  X-Forwarded-Groups \
                  X-Forwarded-Name \
                  X-Forwarded-Email \
                  X-Forwarded-Provider \
                  X-Forwarded-Subject
  }

  reverse_proxy whoami:80
}
```

offa/config.yaml

```
server:

logging:
  access:
    stderr: true
  internal:
    level: info
    stderr: true

sessions:
  ttl: 3600
  cookie_domain: example.com

auth:
  - domain: whoami.example.com
    require:
      groups: users

federation:
  entity_id: https://offa.example.com
  trust_anchors:
    - entity_id: https://ta.example.com
  authority_hints:
    - https://ta.example.com
  logo_uri: https://offa.example.com/static/img/offa-
  key_storage: /data
  use_resolve_endpoint: true
  use_entity_collection_endpoint: true
```

OFFA - Openid Federation Forward Auth

Demo: <https://hello.test.fedcloud.eu>

```
Hostname: 4aa7f0bc4746
IP: 127.0.0.1
IP: ::1
IP: 172.18.0.14
RemoteAddr: 172.18.0.9:60850
GET / HTTP/1.1
Host: hello.test.fedcloud.eu
User-Agent: Mozilla/5.0 ...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.5
Cookie: offa-session=SLn7J9WOST...
...
Via: 2.0 Caddy
X-Forwarded-For: XX.XX.XXX.XXX
X-Forwarded-Host: hello.test.fedcloud.eu
X-Forwarded-Proto: https
X-Forwarded-Provider: https://idp.mivanci.incubator.hexaa.eu
X-Forwarded-Subject: testuserid
X-Forwarded-User: testuserid
```

Openid
Federation
Forward
Auth





Openid
Federation
Forward
Auth

