Authentication Section

A detailed look at Delegated Authentication options available and setup examples.

Config:

> This is a new section introduced in LTS 24.10 which replaces the previous Delegated Authentication options found within the Synapse section. Your previous configuration will be upgraded on first-run of the newer LTS.

In the Authentication section you will find options to configure settings specific to Authentication. Regardless of if you are using the Matrix Authentication Server, or have enabled Legacy Auth, the settings on this page will remain the same.

However please note, MAS does not support delegated authentication with SAML or GroupSync - if you wish to enable either of these you will need to return to the Host section and enable Legacy Auth.

All settings configured via the UI in this section will be saved to your `deployment.yml`, with the contents of secrets being saved to `secrets.yml`. You will find specific configuration examples in each section.

Config Example

- `deployment.yml`

```
metadata:
annotations:
  ui.element.io/layer: |
    components:
      synapse:
spec:
  components:
    synapse:
      config:
        delegatedAuth:
```

- `secrets.yml`

```
kind: Secret
metadata:
```

```
  name: synapse
  namespace: element-onprem
data:
```

By default, if you do not change any settings on this page, defaults will be added to your configuration file/s (see example below).

Config Example

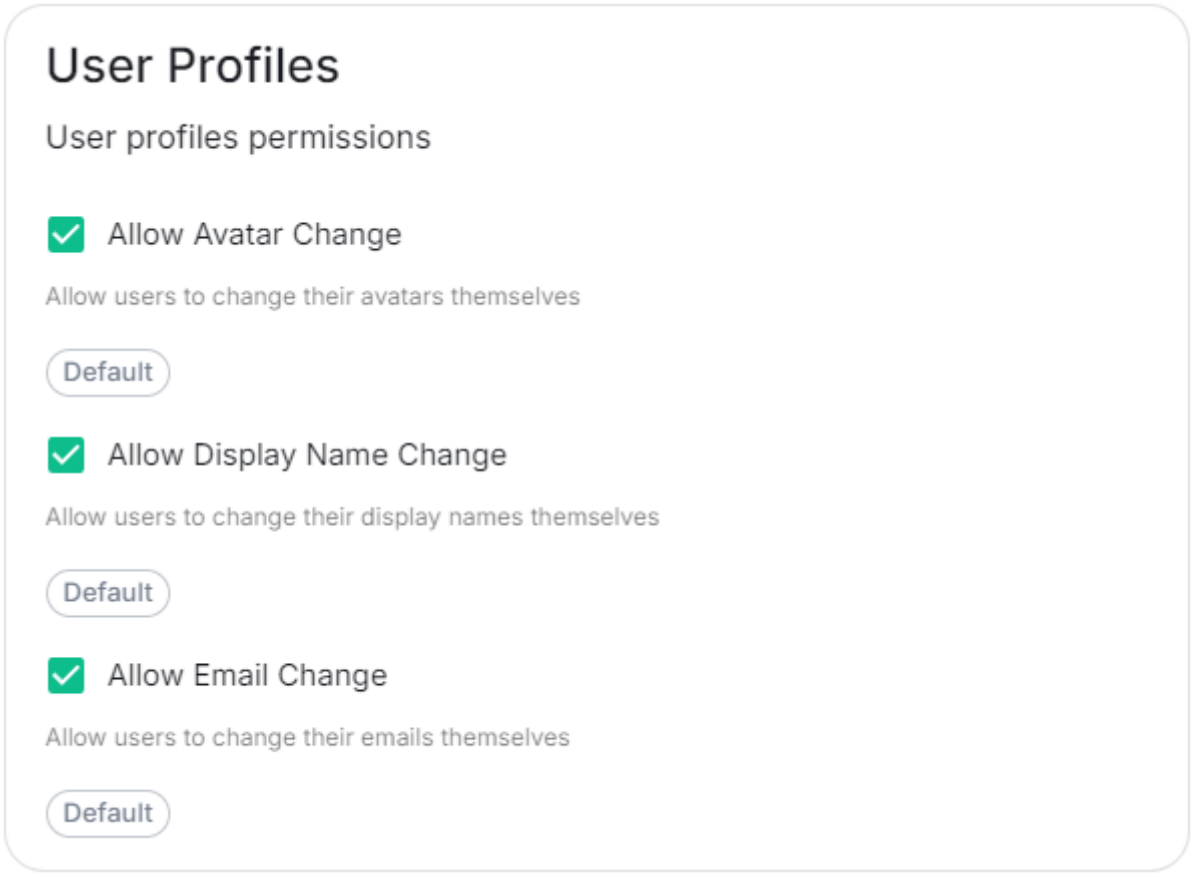
- deployment.yml

```
metadata:
  annotations:
    ui.element.io/layer: |
      components:

spec:
  synapse:
    config:
      delegatedAuth:
        localPasswordDatabase:
          enableRegistration: false # Note, if you deploy without any authentication methods
enabled, the installer will default to Local Accounts.
```

- secrets.yml

```
apiVersion: v1
kind: Secret
metadata:

data:
  ldapBindPassword: examplePassword
```

User Profiles



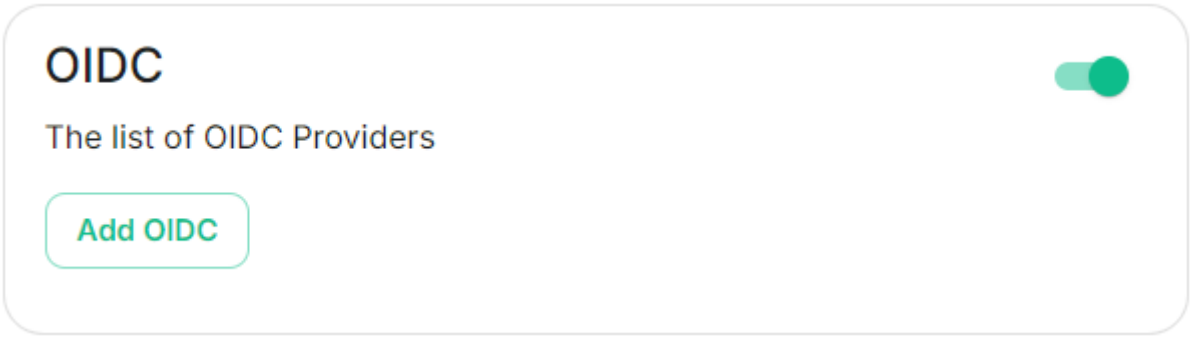
Config Example

```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
          userProfiles:
            allowAvatarChange: true # Not present if left as default
            allowDisplayNameChange: true # Not present if left as default
            allowEmailChange: true # Not present if left as default
```

The User Profiles section provides some self-explanatory config options to adjust what changes users are allowed to make to their User Profile, such as changing their Display Name. You may wish to restrict this if you'd prefer to delegate the setting of these values to the associated Identity Provider.

OIDC

You can add and configure one, or multiple, OIDC providers - to do so you will need to click the `Add OIDC` / `Add more OIDC` button found after toggling on the ODIC section:

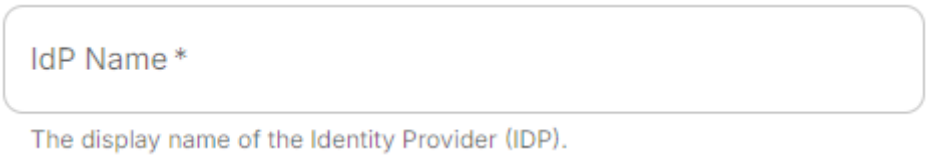


Once an OIDC provider is added, you can remove any providers by clicking the rubbish bin icon found to the left of the provider.

Config Example

```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
          oidc:
            -
```

IdP Name

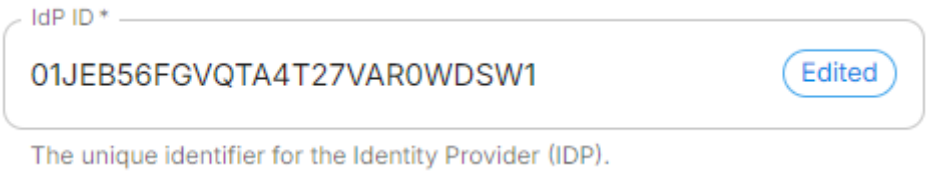


Config Example

```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
```

```
      oidc:
        idpName: example_name # Required
```

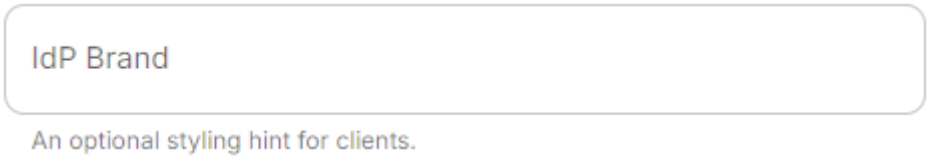
IdP ID



Config Example

```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
          oidc:
            idpId: 01JDS2WKNYTQS21GFAKM9AKD9R # Required
```

IdP Brand



Config Example

```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
```

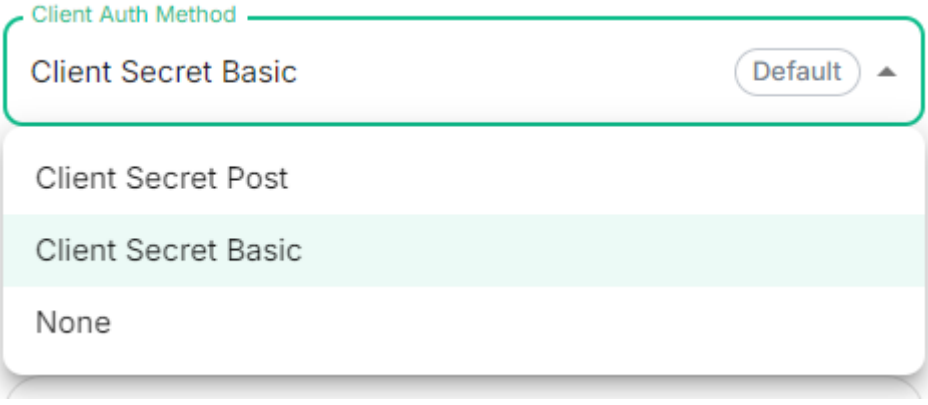
```
    oidc:
      idpBrand: example_brand
```

Issuer

Config Example

```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
          oidc:
            issuer: https://issuer.example.com/ # Required
```

Client Auth Method



Config Example

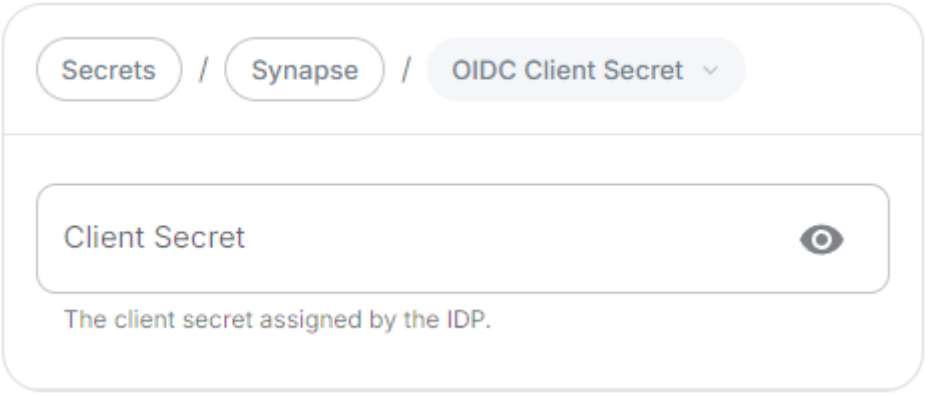
```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
          oidc:
            clientAuthMethod: client_secret_basic # If no `clientAuthMethod` defined, will default to
`client_secret_basic`
            # clientAuthMethod: client_secret_post
            # clientAuthMethod: none
```

Client ID

Config Example

```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
          oidc:
            clientId: example_client_id
```

Client Secret



Config Example

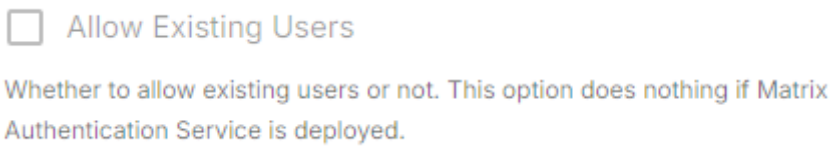
- deployment.yml

```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
          oidc:
            clientSecretSecretKey: oidcClientSecret
```

- secrets.yml

```
apiVersion: v1
kind: Secret
metadata:
  name: synapse
  namespace: element-onprem
data:
  oidcClientSecret: U2VjdXJlT0lEQ0NsaWVudFNlY3JldA==
```

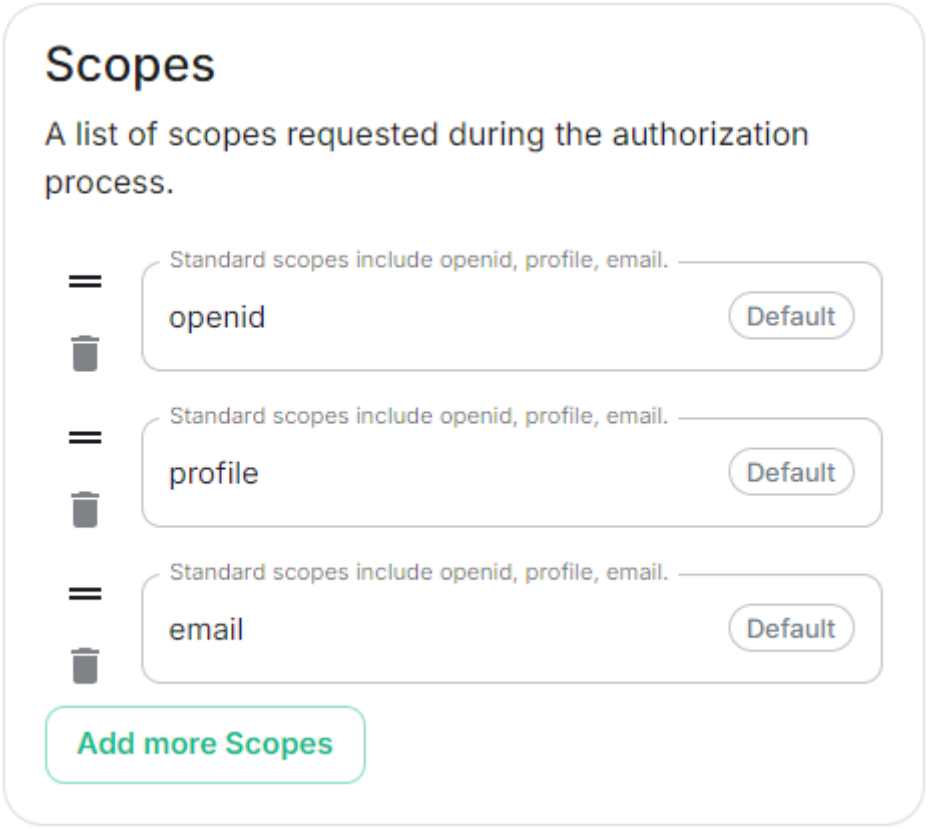
Allow Existing Users



Config Example

```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
          oidc:
```

Scopes



Config Example

```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
          oidc:
```

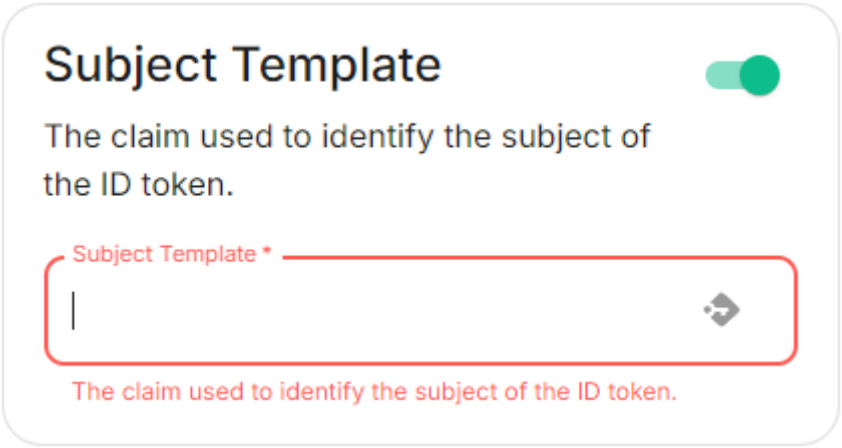
```
      scopes:
        - openid
        - profile
        - email
```

User Mapping Provider

Config Example

```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
          oidc:
            userMappingProvider:
```

Subject Template

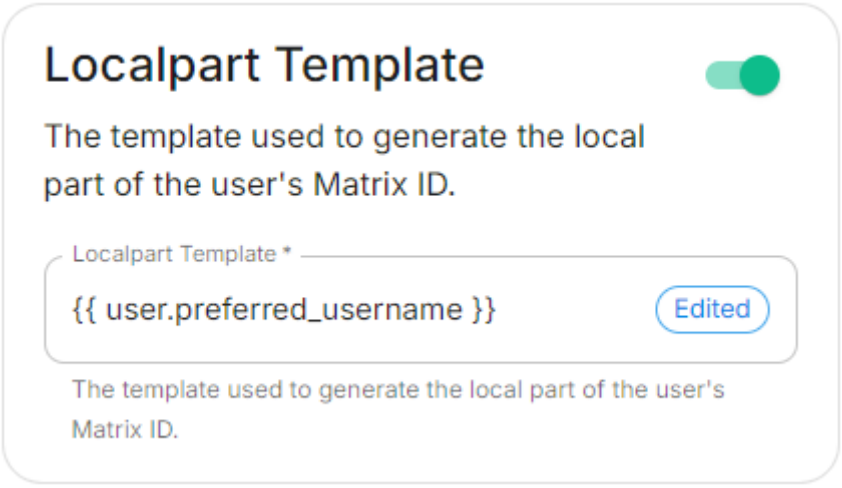


Config Example

```
spec:
  components:
    synapse:
```

```
    config:
      delegatedAuth:
        oidc:
          userMappingProvider:
            subjectTemplate: '{{ user.subject }}'
```

Localpart Template



Config Example

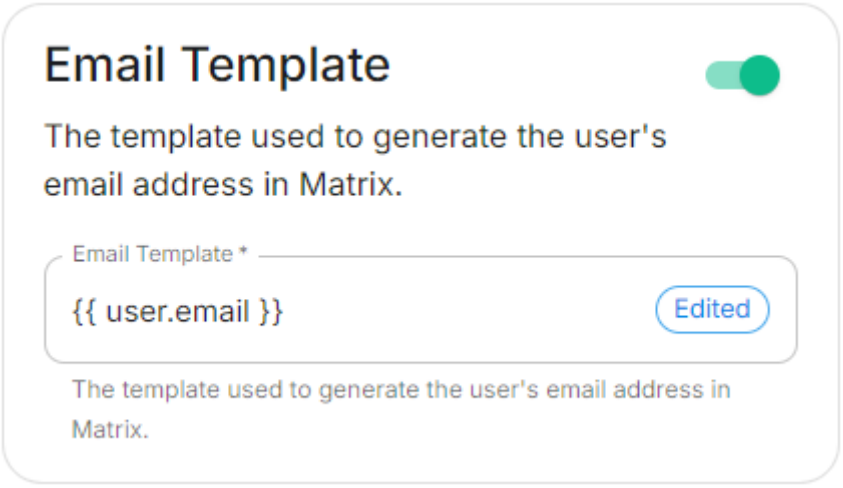
```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
          oidc:
            userMappingProvider:
              localpartTemplate: '{{ user.preferred_username }}'
```

Display Name Template

Config Example

```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
          oidc:
            userMappingProvider:
              displayNameTemplate: '{{ user.name }}'
```

Email Template

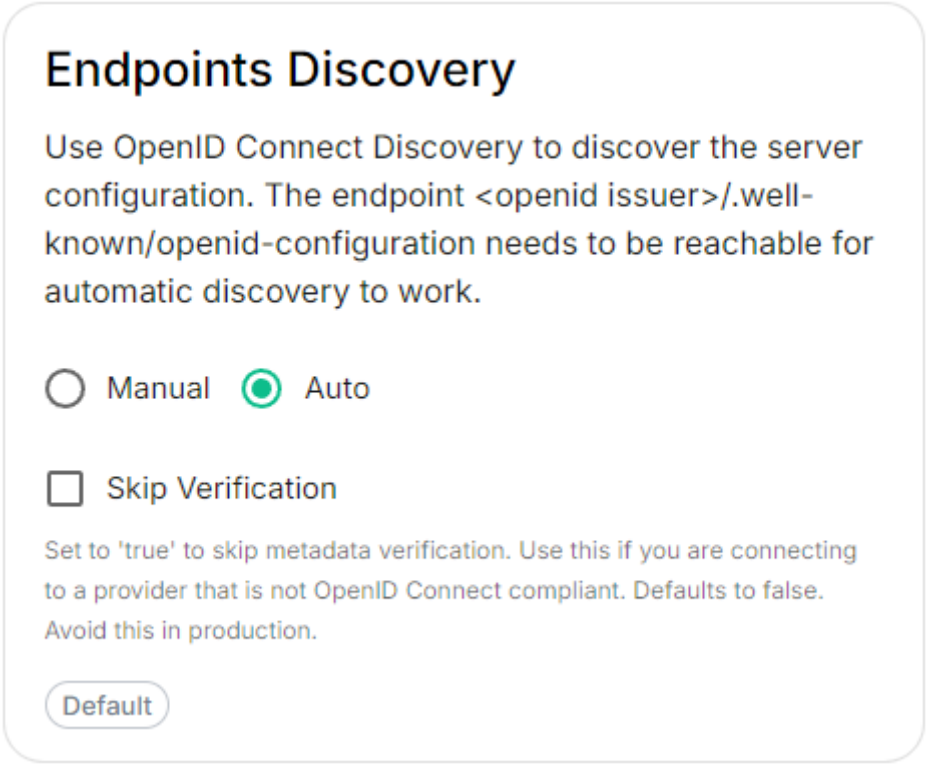


Config Example

```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
          oidc:
            userMappingProvider:
              emailTemplate: '{{ user.email }}'
```

Endpoints Discovery

Auto Discovery



Config Example

```
spec:
  components:
    synapse:
```

```
    config:
      delegatedAuth:
        oidc:
          - clientId: synapsekieranml
            clientSecretSecretKey: oidcClientSecret
            endpointsDiscovery:
              skipVerification: false
            idpId: 01JDS2WKNYTQS21GFAKM9AKD9R
            idpName: Keycloak
            issuer: https://keycloak.ems-support.element.dev/realms/matrix
            scopes:
              - openid
              - profile
              - email
            userMappingProvider:
              displayNameTemplate: '{{ user.name }}'
              emailTemplate: '{{ user.email }}'
```

Skip Verification

Config Example

```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
          oidc:
            - clientId: synapsekieranml
              clientSecretSecretKey: oidcClientSecret
              endpointsDiscovery:
                skipVerification: false
              idpId: 01JDS2WKNYTQS21GFAKM9AKD9R
              idpName: Keycloak
              issuer: https://keycloak.ems-support.element.dev/realms/matrix
              scopes:
```

```
      - openid
      - profile
      - email
    userMappingProvider:
      displayNameTemplate: '{{ user.name }}'
      emailTemplate: '{{ user.email }}'
```

Backchannel Logout Enabled

> The Matrix Authentication Service does not support configuring Backchannel Logout. You can only configure Backchannel logout if you have enabled `Legacy Auth` from the Host Section.

☐ Backchannel Logout Enabled

An optional flag to enable/disable backchannel logout support.

Config Example

```
spec:
  components:
    synapse:
      config:
        delegatedAuth:
          oidc:
            - clientId: synapsekieranml
              clientSecretSecretKey: oidcClientSecret
              endpointsDiscovery:
                skipVerification: false
              idpId: 01JDS2WKNYTQS21GFAKM9AKD9R
              idpName: Keycloak
              issuer: https://keycloak.ems-support.element.dev/realms/matrix
              scopes:
                - openid
                - profile
                - email
```

```
    userMappingProvider:
      displayNameTemplate: '{{ user.name }}'
      emailTemplate: '{{ user.email }}'
```

SAML

> The Matrix Authentication Service does not support SAML and it is recommended to switch to OIDC. You can only enable SAML authentication if you have enabled `Legacy Auth` from the Host Section.

LDAP

Local Accounts

Revision #18
Created 6 November 2024 10:22:14 by Kieran Mitchell Lane
Updated 5 December 2024 15:31:56 by Kieran Mitchell Lane