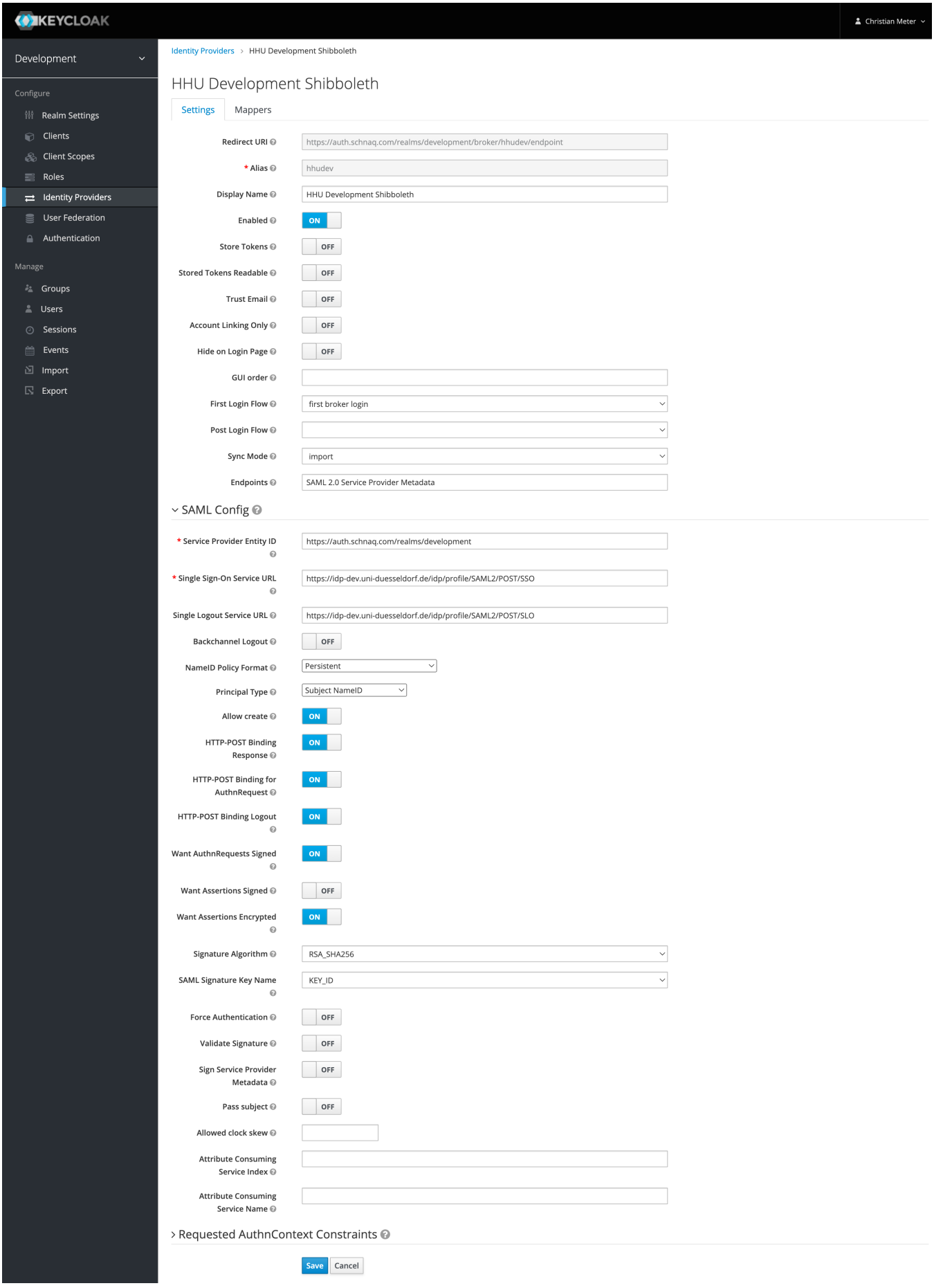
Admins - Keycloak an Shibboleth IdP anbinden

Hier ist beschrieben, wie Keycloak zu konfigurieren ist, damit eine SAML-Authentifizierung über den Shibboleth Identity Provider möglich ist. In den folgenden Screenshots wird der Development-Server von schnaq mit dem Development-IdP der HHU verbunden.

Eine detailliertere und weiterreichende Dokumentation findet sich unter: https://www.blog.albert-stark.de/p/shibboleth-and-keycloak/


KEYCLOAK
Christian Meter
Development
Configure
Realm Settings
Clients
Client Scopes
Roles
Identity Providers
User Federation
Authentication
Manage
Groups
Users
Sessions
Events
Import
Export
Identity Providers › HHU Development Shibboleth
HHU Development Shibboleth
Settings
Mappers
Redirect URI
https://auth.schnaq.com/realms/development/broker/hhudev/endpoint
* Alias
hhudev
Display Name
HHU Development Shibboleth
Enabled
ON
Store Tokens
OFF
Stored Tokens Readable
OFF
Trust Email
OFF
Account Linking Only
OFF
Hide on Login Page
OFF
GUI order
First Login Flow
first broker login
Post Login Flow
Sync Mode
import
Endpoints
SAML 2.0 Service Provider Metadata
SAML Config
* Service Provider Entity ID
https://auth.schnaq.com/realms/development
* Single Sign-On Service URL
https://idp-dev.uni-duesseldorf.de/idp/profile/SAML2/POST/SSO
Single Logout Service URL
https://idp-dev.uni-duesseldorf.de/idp/profile/SAML2/POST/SLO
Backchannel Logout
OFF
NameID Policy Format
Persistent
Principal Type
Subject NameID
Allow create
ON
HTTP-POST Binding Response
ON
HTTP-POST Binding for AuthnRequest
ON
HTTP-POST Binding Logout
ON
Want AuthnRequests Signed
ON
Want Assertions Signed
OFF
Want Assertions Encrypted
ON
Signature Algorithm
RSA_SHA256
SAML Signature Key Name
KEY_ID
Force Authentication
OFF
Validate Signature
OFF
Sign Service Provider Metadata
OFF
Pass subject
OFF
Allowed clock skew
Attribute Consuming Service Index
Attribute Consuming Service Name
Requested AuthnContext Constraints
Save
Cancel

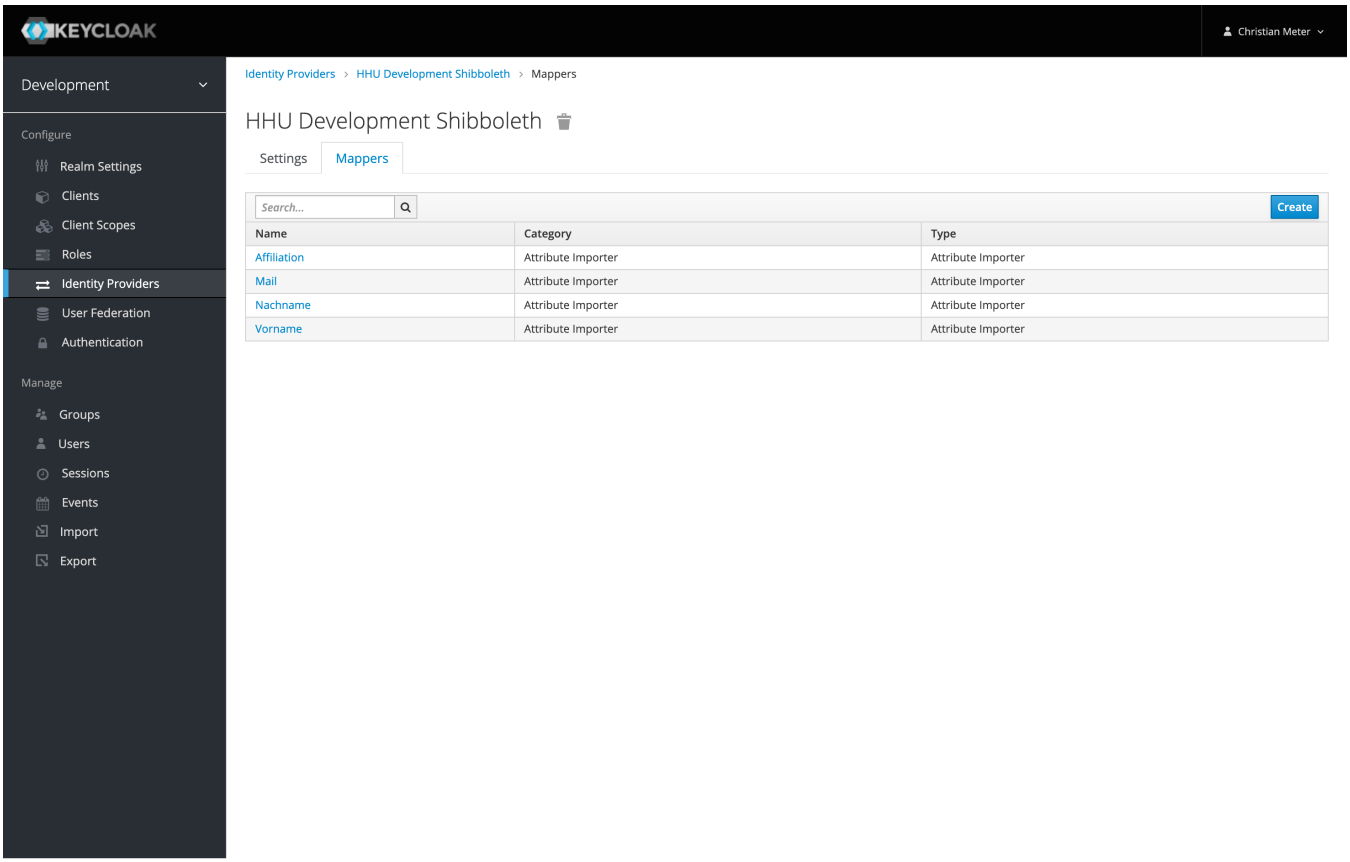
Die resultierenden Metadaten müssen dem IdP zur Verfügung gestellt werden (diese sind sehr minimalistisch, falls jemand weiß, wie Zusatzinformationen wie `Organization`, `ContactPerson`, `AttributeConsumingService` etc angegeben werden können, bitte melden). In unserem Fall wird die generierte xml-Datei per Ansible auf dem IdP abgelegt und per `FilesystemMetadataProvider` in der `metadata-providers.xml` integriert.

Attributfreigabe im IdP:

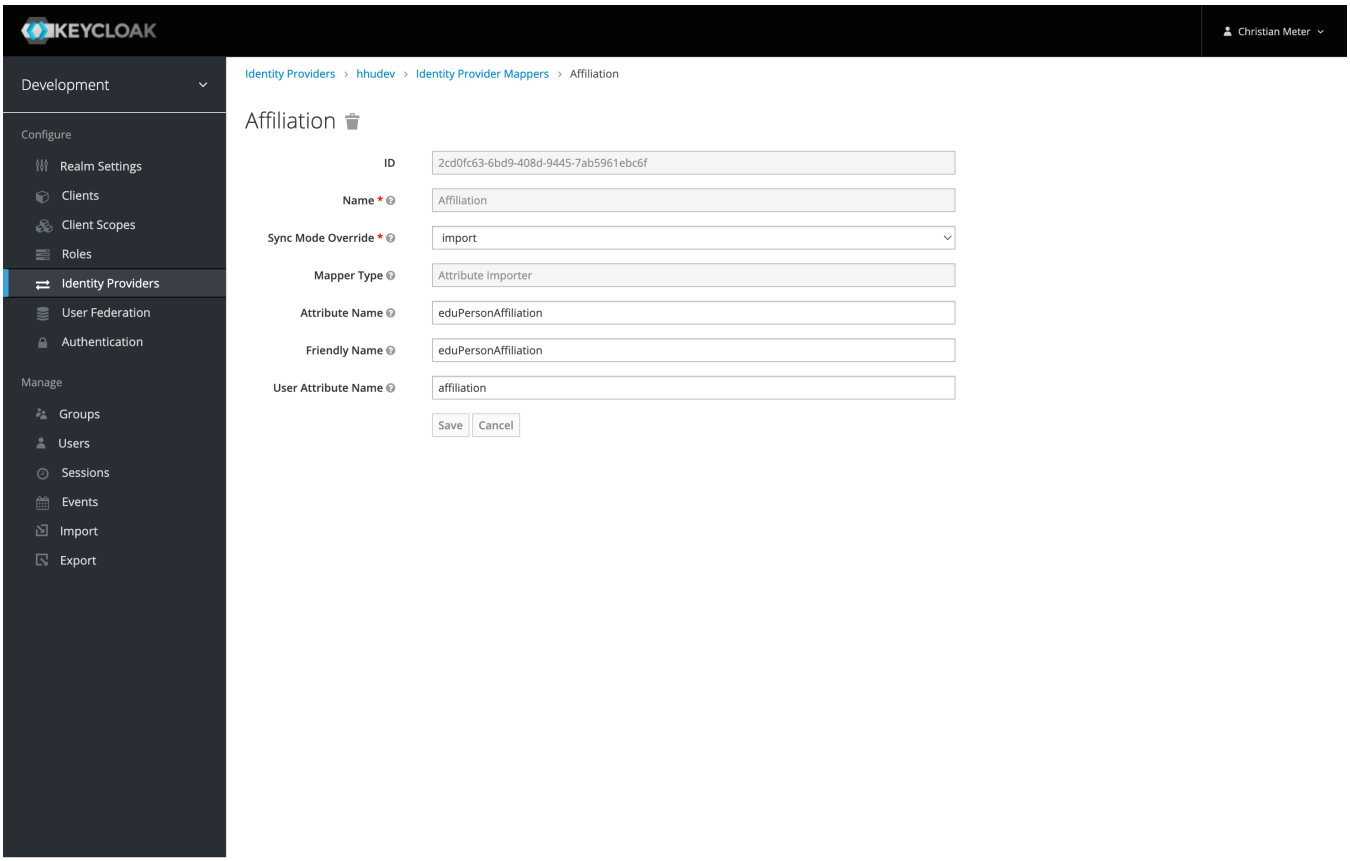
IdP - attribute-resolver.xml

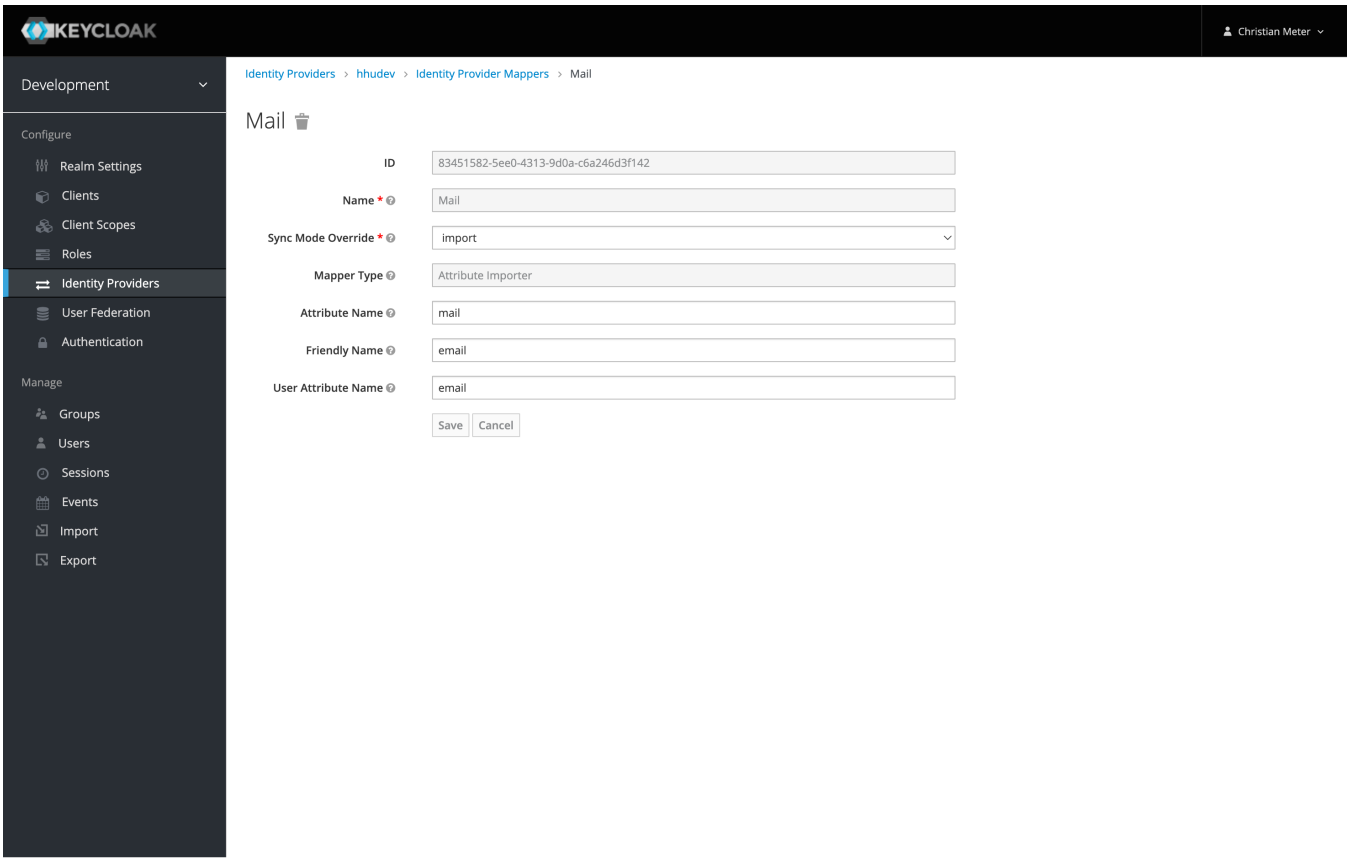
```
<AttributeFilterPolicy id="schnaq">
   <PolicyRequirementRule xsi:type="Requester" value="https://auth.schnaq.com/realms/schnaq" />
   <AttributeRule attributeID="mail"                permitAny="true" />
   <AttributeRule attributeID="givenName"           permitAny="true" />
   <AttributeRule attributeID="sn"                  permitAny="true" />
   <AttributeRule attributeID="displayName"         permitAny="true" />
   <AttributeRule attributeID="eduPersonAffiliation">
      <PermitValueRule xsi:type="Value" value="employee" caseSensitive="false" />
   </AttributeRule>
</AttributeFilterPolicy>
```

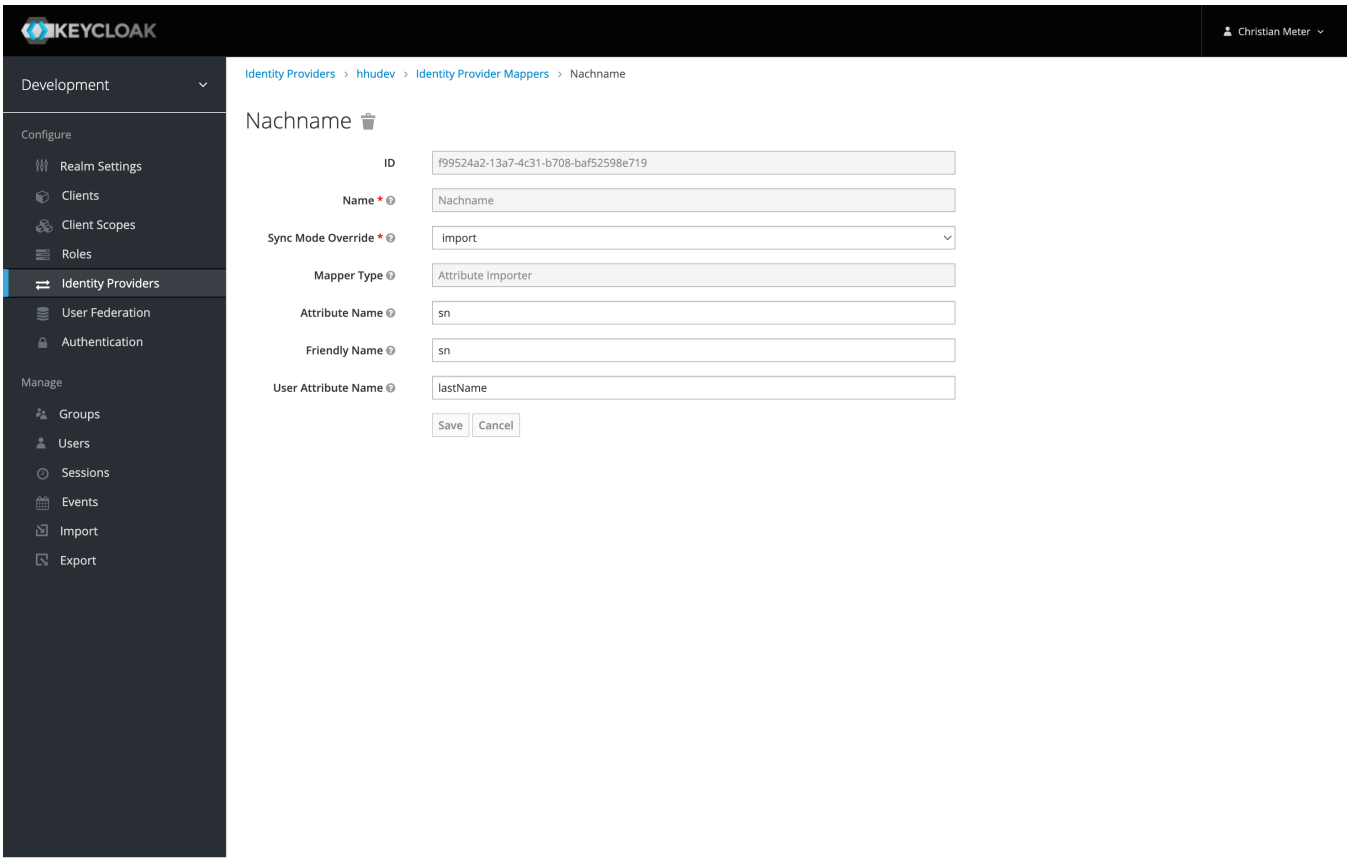
`displayName` wird vermutlich in Zukunft genutzt. eduPersonAffiliation wird genutzt um Angestellten ("employee") zusätzliche Dienste zur Verfügung zu stellen (s. letzter Screenshot). Über Keycloak-Mapper werden die Werte der SAML-Attribute dem Dienst verfügbar gemacht (die Zuordnung der User passiert vermutlich implizit über die `persistentId`):

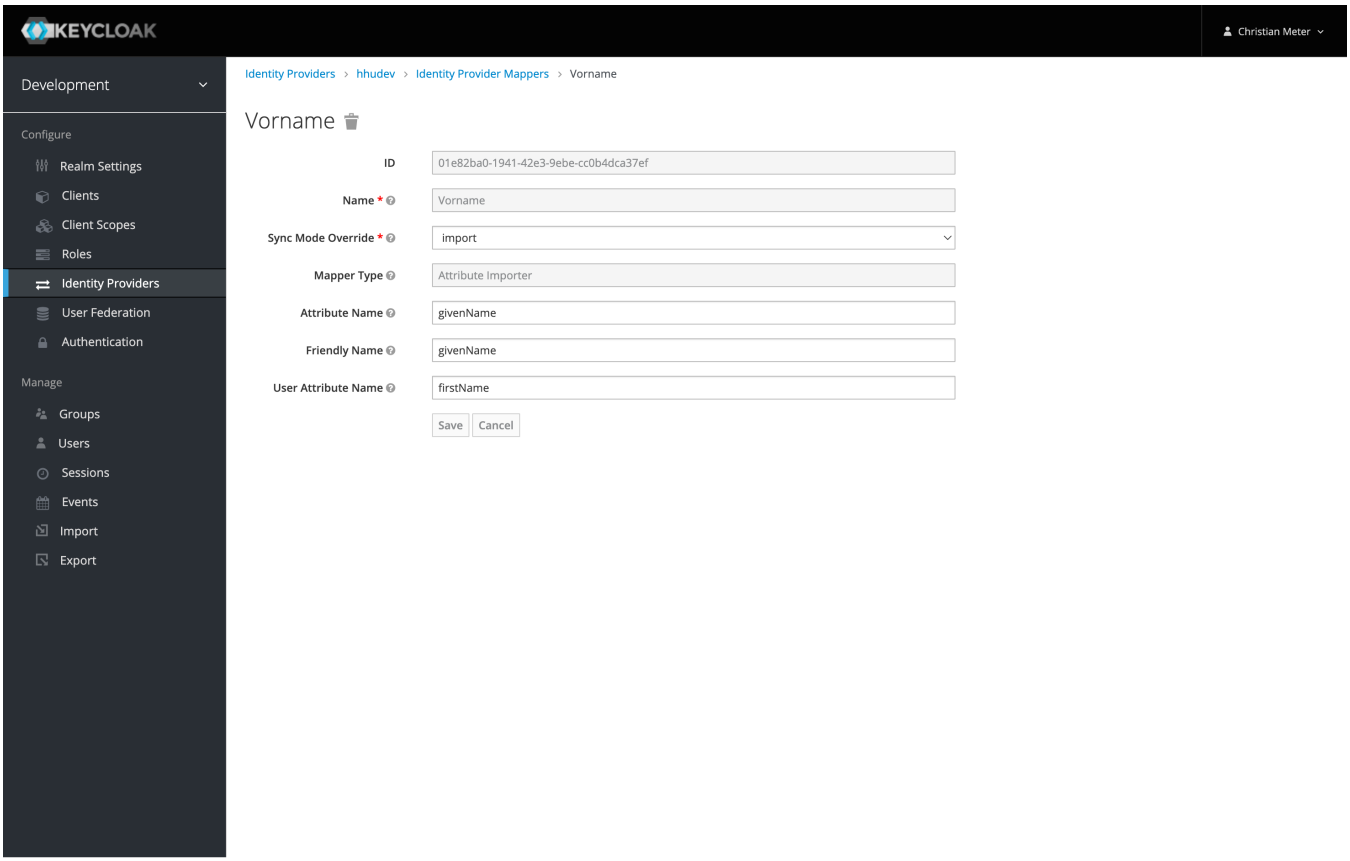


Die einzelnen Mapper: "Attribute Name" muss genau der Bezeichnung der SAML-Assertion entsprechen (oder URN), "Attribute User Name" dem Zielbezeichner.

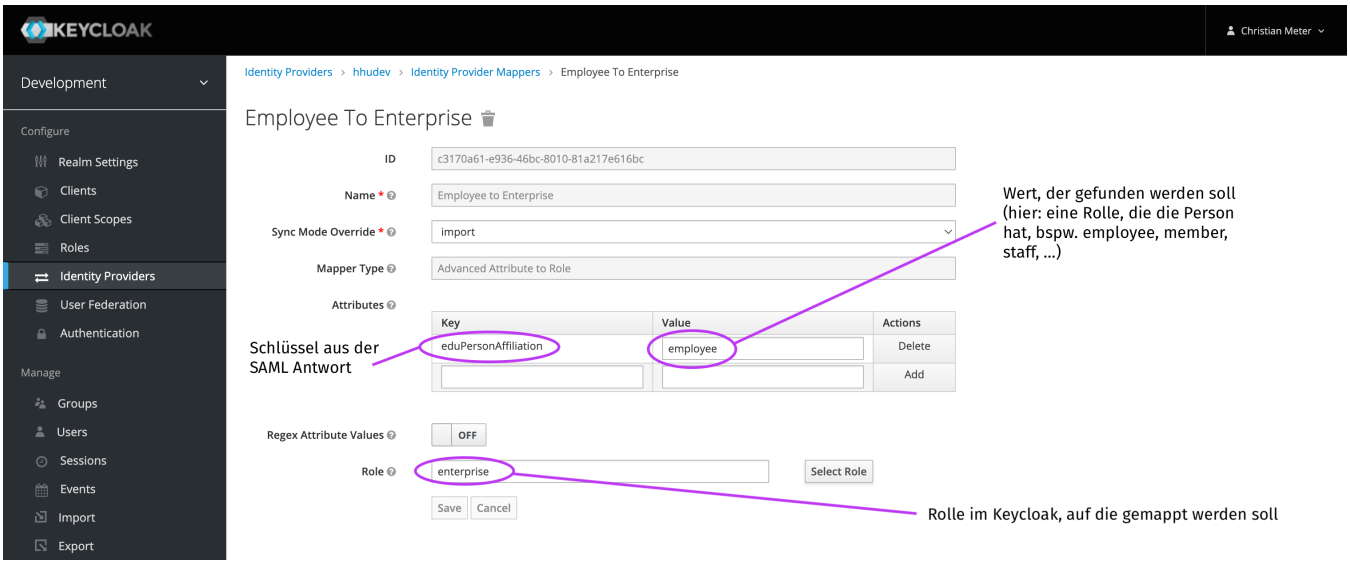

KEYCLOAK
Christian Meter
Development
Configure
Realm Settings
Clients
Client Scopes
Roles
Identity Providers
User Federation
Authentication
Manage
Groups
Users
Sessions
Events
Import
Export
Identity Providers › hhudev › Identity Provider Mappers › Affiliation
Affiliation
ID
2cd0fc63-6bd9-408d-9445-7ab5961ebc6f
Name *
Affiliation
Sync Mode Override *
import
Mapper Type
Attribute Importer
Attribute Name
eduPersonAffiliation
Friendly Name
eduPersonAffiliation
User Attribute Name
affiliation
Save
Cancel


KEYCLOAK
Christian Meter
Development
Configure
Realm Settings
Clients
Client Scopes
Roles
Identity Providers
User Federation
Authentication
Manage
Groups
Users
Sessions
Events
Import
Export
Identity Providers › hhudev › Identity Provider Mappers › Mail
Mail
ID
83451582-5ee0-4313-9d0a-c6a246d3f142
Name *
Mail
Sync Mode Override *
import
Mapper Type
Attribute Importer
Attribute Name
mail
Friendly Name
email
User Attribute Name
email
Save
Cancel


KEYCLOAK
Christian Meter
Development
Configure
Realm Settings
Clients
Client Scopes
Roles
Identity Providers
User Federation
Authentication
Manage
Groups
Users
Sessions
Events
Import
Export
Identity Providers › hhudev › Identity Provider Mappers › Nachname
Nachname
ID
f99524a2-13a7-4c31-b708-baf52598e719
Name *
Nachname
Sync Mode Override *
import
Mapper Type
Attribute Importer
Attribute Name
sn
Friendly Name
sn
User Attribute Name
lastName
Save
Cancel


KEYCLOAK
Christian Meter
Development
Configure
Realm Settings
Clients
Client Scopes
Roles
Identity Providers
User Federation
Authentication
Manage
Groups
Users
Sessions
Events
Import
Export
Identity Providers › hhudev › Identity Provider Mappers › Vorname
Vorname
ID
01e82ba0-1941-42e3-9ebe-cc0b4dca37ef
Name *
Vorname
Sync Mode Override *
import
Mapper Type
Attribute Importer
Attribute Name
givenName
Friendly Name
givenName
User Attribute Name
firstName
Save
Cancel

Angestellte bekommen Zugang zur Enterprise-Version:



Bei Fragen:

idm ät hhu Punkt de

christian ät schnaq Punkt com