didmos2 Core

Introduction

didmos2 Core is a django based web-backend written in python for identity management which can be deployed inside a Docker environment or as a debian package. The backend uses an OpenLdap directory as database and provides a Restful ScimV2 API as an interface. Over this API it is possible to create new resources like users, groups and other objects with predefined processes inside the metadirectory. Modification and deletion of such objects is also possible as well as implementation of complex workflows to provide asynchronous execution of modifications as well as e.g. notifications or execution of any other python code. From the beginning didmos2 Core was developed to provide a high amount of flexibility and extensibility.

Apps

The backend is structured in Apps which provide different functionality.

LDAP-App

This app handles all requests against the metadirectory. It bundles requests where needed, caches data to avoid redundant data requests, implements several OpenLDAP server controls and many other things.

A strength of this app is the dynamic building of almost all needed LDAP filters. This allows the developer to use complex LDAP filter constructs even without deeper knowledge in LDAP search filters. If necessary the app also provides many interceptor entry points, where modifications prior or after sending the request to the metadirectory is possible.

PDP-App

All requests in didmos2 Core are going through an authorisation process to check, whether the requestor has sufficient rights for this request. The PDP App is mainly a wrapper around the didmos-RBAC library which is a python implementation of role based access control. The library is capable of managing users, roles, permissions and sessions.

Further information about the didmos-RBAC libary can be found here.

Task-App

Not all accruing jobs inside an IdM can be done synchronously. This is especially true for every interaction of the IdM with other sources, let it be users or other servers. To cover all possible cases in which asynchronous are necessary, tasks are implemented quite generically in didmos2 Core. This allows to implement the concept of user-requests with the same mechanism as lifecycle events or automatic role assignment.

Customer-App

This App is not directly part of didmos2 Core. But allows to override almost all functions used inside didmos2 Core. This gives developers the freedom to both, relay on well tested functionality inside the core as well as the flexibility to implement own functions or even full custom endpoints.

didmos2 Core has a ready to use default configuration which allows to directly start working with didmos2. But since this might not fit for every project, all configuration parameters can be overwritten inside the customer App. This works in addition to your configuration implementation let it be configuration files or the usage of the didmos2-configserver.

OpenLDAP as metadirectory

didmos2 Core is reliant on an OpenLDAP directory server as metadirectory. The backend is highly flexible regarding object classes, structure and attribute naming. Nevertheless some functionality, the backend relies on is provided by OpenLDAP overlays which are kind of extensions to the default OpenLDAP implementation which makes them necessary for the metadirectory to have them installed.

Configuration parameters

A complete list of all configuration parameters can be found here.

Api documentation

- ScimV2 API documentation
- PDP API documentation