Configuration report
PRIAM implementation guide

Sommaire

Introduction

This document is an explanatory guide detailing the implementation of the PRIAM solution by the service provider. It includes essential information on the configuration, environment variables and specific routes that the provider needs to implement to fully integrate PRIAM into his system. The various sections of the document cover key points such as retrieving data access rights, manipulating data values, rectifying and deleting data, as well as authentication-related functionalities. Each route is accompanied by details of its descriptions, required parameters, and expected returns, making it easy to understand and seamlessly integrate PRIAM into the supplier's infrastructure.

Configuration

Environment variables

Within each Spring Boot service (except Eureka), it is necessary to add these variables in application.properties :
- spring.security.oauth2.client.registration.provider-client.client-id :
  - L'id enregistré dans la configuration OAuth2.
- spring.security.oauth2.client.registration.provider-client.client-secret :
  - Le secret enregistré dans la configuration OAuth2.
- spring.security.oauth2.client.registration.provider-client.authorization-grant-type :
  - Actuellement, seul *authorization_code* est pris en compte.
- spring.security.oauth2.client.provider.provider-client.authorization-uri :
  - L'uri permettant l'autorisation.
  - Exemple : http://localhost:9090/oauth/authorize
- spring.security.oauth2.client.provider.provider-client.token-uri :
  - L'uri permettant de récupérer le token.
  - Exemple : http://localhost:9090/oauth/token
- spring.security.oauth2.client.provider.provider-client.user-info-uri :
  - L'uri pour récupérer les infos de l'utilisateur.
  - Exemple : http://localhost:9090/userinfo
- spring.security.oauth2.client.provider.provider-client.user-name-attribute :
  - L'attribut dans la réponse de l'utilisateur fournissant son nom d'utilisateur.
- spring.security.oauth2.client.registration.provider-client.redirect-uri :
  - L'URI vers laquelle l'utilisateur sera redirigé après une autorisation réussie sur le fournisseur OAuth2.

Methods to implement in the Provider service

GET – <host>/api/dataAccessRight

Service method to be implemented: getPersonalDataValues

Description

Retrieves the values of several data items based on the reference identifier, the dataType concerned, the data names of the dataType, and the primary keys with their values to identify which rows are concerned.

Paramètres

- idRef : String
- dataTypeName : String
- attributes: List<String>

Retour

List<Map<String, String>>

POST – <host>/api/dataValue

Service method to be implemented: getDataValue

Description

Retrieves the value of a single data item based on the reference identifier, dataType, data name, and primary keys with their value to identify which row is concerned.

Parameters

Body (json) :
{
      idRef: String,
      dataName: String,
      primaryKeys: Map<String, String> (key : nom de l'attribut, value: sa valeur)
}

Retour

Map<value: String, String>

POST - <host>/api/rectification

Méthode du service à implémenter : Rectification

Description

Used when a rectification request has been accepted by an administrator. It will modify the value of the data concerned.

Parameters

Body (json) :
```
{
        idRef: String,
        dataName: String,
        dataTypeName: String,
        newValue: String,
        primaryKeys: Map<String, String> (key : nom de l'attribut, value: sa valeur)
}
```

Retour

Void

POST - <host>/api/forgotten

Service method to be implemented: Erasure

Description

Used when a request for the right to be forgotten has been accepted by an administrator. It will delete the value of the data concerned.

Parameters

Body (json) :
```
{
```

idRef: String,
        dataName: String,
        dataTypeName: String,
        primaryKeys: Map<String, String> (key : nom de l'attribut, value: sa valeur)
}

Retour

Void