Policy Enforcement Point Interface Control Document

EOEPCA.ICD.xxx

TVUK System Team

Version 1.0, 30/11/2020:

Policy Enforcement Point Interface Control Document

EO Exploitation Platform Common Architecture

Policy Enforcement Point Interface Control Document

EOEPCA.ICD.xxx

COMMENTS and ISSUES	**PDF**
If you would like to raise comments or issues on this document, please do so by raising an Issue at the following URL https://github.com/EOEPCA/um-pep-engine/issues.	This document is available in PDF format here.
EUROPEAN SPACE AGENCY CONTRACT REPORT The work described in this report was done under ESA contract. Responsibility for the contents resides in the author or organisation that prepared it.	**TELESPAZIO VEGA UK Ltd** 350 Capability Green, Luton, Bedfordshire, LU1 3LU, United Kingdom. Tel: +44 (0)1582 399000 www.telespazio-vega.com

AMENDMENT HISTORY

This document shall be amended by releasing a new edition of the document in its entirety. The Amendment Record Sheet below records the history and issue status of this document.

Table 1. Amendment Record Sheet

ISSUE	DATE	REASON
0.1	dd/mm/yyyy	Initial in-progress draft

Chapter 1. Introduction

1.1. Purpose and Scope

This document presents the Policy Enforcement Point Interfaces for the Common Architecture. It servers as a complementary document to its corresponding Software Design Document.

Chapter 2. Overview

This Interface Control Document (ICD) is a companion to the System Design Document for the Policy Enforcement Point. The ICD provides a Building Block level specification of the interfaces exposed by the PEP to the rest of EOEPCA components.

Section [Interfaces]

Provides the interface specification of the Building Block.

Chapter 3. Policy Enforcement Point Interfaces

Abstract

*This OpenAPI Document describes the endpoints exposed by Policy Enforcement Point Building Block deployments.

 Using this API will allow to register resources that can be protected using both the Login Service and the Policy Decision Point and access them through the Policy Enforcement Endpoint.

 As an example this documentation uses \"proxy\" as the configured base URL for Policy Enforcement, but this can be manipulated through configuration parameters.*

3.1. Endpoints

3.1.1. PolicyEnforcement

3.1.1.1. Proxy DELETE

`DELETE /proxy/{path}`

Request to Back-End Service

3.1.1.1.1. Description

This operation propagates all headers

3.1.1.1.2. Parameters

Path Parameters

Name	Description	Required
path	Path to the Back-End Service	X

Header Parameters

Name	Description	Required
Authorization	RPT Token generated through UMA Flow	-

3.1.1.1.3. Return Type

-

3.1.1.1.4. Responses

Table 2. http response codes

Code	Message	Datatype
200	OK	<<>>
401	Unauthorized access request.	<<>>

3.1.1.1.5. Samples

3.1.1.2. Proxy GET

`GET /proxy/{path}`

Request to Back-End Service

3.1.1.2.1. Description

This operation propagates all headers and query parameters

3.1.1.2.2. Parameters

Path Parameters

Name	Description	Required
path	Path to the Back-End Service	X

Header Parameters

Name	Description	Required
Authorization	RPT Token generated through UMA Flow	-

3.1.1.2.3. Return Type

-

3.1.1.2.4. Responses

Table 3. http response codes

Code	Message	Datatype
200	OK	<<>>
401	Unauthorized access request.	<<>>

3.1.1.2.5. Samples

3.1.1.3. Proxy HEAD

`HEAD /proxy/{path}`

Request to Back-End Service

3.1.1.3.1. Description

This operation propagates all headers and query parameters

3.1.1.3.2. Parameters

Path Parameters

Name	Description	Required
path	Path to the Back-End Service	X

Header Parameters

Name	Description	Required
Authorization	RPT Token generated through UMA Flow	-

3.1.1.3.3. Return Type

-

3.1.1.3.4. Responses

Table 4. http response codes

Code	Message	Datatype
200	OK	<<>>
401	Unauthorized access request.	<<>>

3.1.1.3.5. Samples

3.1.1.4. Proxy POST

`POST /proxy/{path}`

Request to Back-End Service

3.1.1.4.1. Description

This operation propagates all headers, query parameters and body

3.1.1.4.2. Parameters

Path Parameters

Name	Description	Required
path	Path to the Back-End Service	X

Header Parameters

Name	Description	Required
Authorization	RPT Token generated through UMA Flow	-

3.1.1.4.3. Return Type

-

3.1.1.4.4. Responses

Table 5. http response codes

Code	Message	Datatype
200	OK	<<>>
401	Unauthorized access request.	<<>>

3.1.1.4.5. Samples

3.1.1.5. Proxy PUT

`PUT /proxy/{path}`

Request to Back-End Service

3.1.1.5.1. Description

This operation propagates all headers, query parameters and body

3.1.1.5.2. Parameters

Path Parameters

Name	Description	Required
path	Path to the Back-End Service	X

Header Parameters

Name	Description	Required
Authorization	RPT Token generated through UMA Flow	-

3.1.1.5.3. Return Type

-

3.1.1.5.4. Responses

Table 6. http response codes

Code	Message	Datatype
200	OK	<<>>
401	Unauthorized access request.	<<>>

3.1.1.5.5. Samples

3.1.1.6. Proxy PATCH

`PATCH /proxy/{path}`

Request to Back-End Service

3.1.1.6.1. Description

This operation propagates all headers, query parameters and body

3.1.1.6.2. Parameters

Path Parameters

Name	Description	Required
path	Path to the Back-End Service	X

Header Parameters

Name	Description	Required
Authorization	RPT Token generated through UMA Flow	-

3.1.1.6.3. Return Type

-

3.1.1.6.4. Responses

Table 7. http response codes

Code	Message	Datatype
200	OK	<<>>
401	Unauthorized access request.	<<>>

3.1.1.6.5. Samples

3.1.2. PolicyAuthorize

3.1.2.1. Authorize DELETE

`DELETE /authorize`

Request to Back-End Service

3.1.2.1.1. Description

This operation propagates all headers

3.1.2.1.2. Parameters

Path Parameters

Name	Description	Required
path	Path to the Back-End Service	X

Header Parameters

Name	Description	Required
Authorization	RPT Token generated through UMA Flow	-

3.1.2.1.3. Return Type

-

3.1.2.1.4. Responses

Table 8. http response codes

Code	Message	Datatype
200	OK	<<>>
401	Unauthorized access request.	<<>>

3.1.2.1.5. Samples

3.1.2.2. Authorize GET

`GET /authorize`

Request to Back-End Service

3.1.2.2.1. Description

This operation propagates all headers and query parameters

3.1.2.2.2. Parameters

Path Parameters

Name	Description	Required
path	Path to the Back-End Service	X

Header Parameters

Name	Description	Required
Authorization	RPT Token generated through UMA Flow	-

3.1.2.2.3. Return Type

-

3.1.2.2.4. Responses

Table 9. http response codes

Code	Message	Datatype
200	OK	<<>>
401	Unauthorized access request.	<<>>

3.1.2.2.5. Samples

3.1.2.3. Authorize HEAD

`HEAD /authorize`

Request to Back-End Service

3.1.2.3.1. Description

This operation propagates all headers and query parameters

3.1.2.3.2. Parameters

Path Parameters

Name	Description	Required
path	Path to the Back-End Service	X

Header Parameters

Name	Description	Required
Authorization	RPT Token generated through UMA Flow	-

3.1.2.3.3. Return Type

-

3.1.2.3.4. Responses

Table 10. http response codes

Code	Message	Datatype
200	OK	<<>>

Code	Message	Datatype
401	Unauthorized access request.	<<>>

3.1.2.3.5. Samples

3.1.2.4. Authorize POST

`POST /authorize`

Request to Back-End Service

3.1.2.4.1. Description

This operation propagates all headers, query parameters and body

3.1.2.4.2. Parameters

Path Parameters

Name	Description	Required
path	Path to the Back-End Service	X

Header Parameters

Name	Description	Required
Authorization	RPT Token generated through UMA Flow	-

3.1.2.4.3. Return Type

-

3.1.2.4.4. Responses

Table 11. http response codes

Code	Message	Datatype
200	OK	<<>>
401	Unauthorized access request.	<<>>

3.1.2.4.5. Samples

3.1.2.5. Authorize PUT

`PUT /authorize`

Request to Back-End Service

3.1.2.5.1. Description

This operation propagates all headers, query parameters and body

3.1.2.5.2. Parameters

Path Parameters

Name	Description	Required
path	Path to the Back-End Service	X

Header Parameters

Name	Description	Required
Authorization	RPT Token generated through UMA Flow	-

3.1.2.5.3. Return Type

-

3.1.2.5.4. Responses

Table 12. http response codes

Code	Message	Datatype
200	OK	<<>>
401	Unauthorized access request.	<<>>

3.1.2.5.5. Samples

3.1.2.6. Authorize PATCH

`PATCH /authorize`

Request to Back-End Service

3.1.2.6.1. Description

This operation propagates all headers, query parameters and body

3.1.2.6.2. Parameters

Path Parameters

Name	Description	Required
path	Path to the Back-End Service	X

Header Parameters

Name	Description	Required
Authorization	RPT Token generated through UMA Flow	-

3.1.2.6.3. Return Type

-

3.1.2.6.4. Responses

Table 13. http response codes

Code	Message	Datatype
200	OK	<<>>
401	Unauthorized access request.	<<>>

3.1.2.6.5. Samples

3.1.3. Resources

3.1.3.1. Resources GET

`GET /resources`

List all owned resources

3.1.3.1.1. Description

This operation lists all resources filtered by ownership ID. Ownership ID is extracted from the OpenID Connect Token

3.1.3.1.2. Parameters

Header Parameters

Name	Description	Required
Authorization	JWT or Bearer Token	-

3.1.3.1.3. Return Type

array[[resource]]

3.1.3.1.4. Content Type

- application/json

3.1.3.1.5. Responses

Table 14. http response codes

Code	Message	Datatype
200	OK	List[[resource]]

3.1.3.1.6. Samples

3.1.3.2. Resources POST

`POST /resources`

Creates a new Resource reference in the Platform

3.1.3.2.1. Description

This operation generates a new resource reference object that can be protected. Ownership ID is set to the unique ID of the End-User

3.1.3.2.2. Parameters

3.1.3.2.3. Body Parameter

Name	Description	Required
NewResource	*NewResource*	X

Header Parameters

Name	Description	Required
Authorization	JWT or Bearer Token	-

3.1.3.2.4. Return Type

[resource]

3.1.3.2.5. Content Type

- application/json

3.1.3.2.6. Responses

Table 15. http response codes

Code	Message	Datatype
200	OK	[resource]
401	UNAUTHORIZED	<<>>
404	NOT FOUND	<<>>

3.1.3.2.7. Samples

3.1.3.3. Resources DELETE

`DELETE /resources/{resource_id}`

Deletes an owned Resource Reference from the Platform

3.1.3.3.1. Description

This operation removes an existing Resource reference owned by the user.

3.1.3.3.2. Parameters

Path Parameters

Name	Description	Required
resource_id	Unique Resource ID	X

Header Parameters

Name	Description	Required
Authorization	JWT or Bearer Token	-

3.1.3.3.3. Return Type

-

3.1.3.3.4. Responses

Table 16. http response codes

Code	Message	Datatype
200	OK	<<>>
401	UNAUTHORIZED	<<>>
404	NOT FOUND	<<>>

3.1.3.3.5. Samples

3.1.3.4. Resource GET (ID)

```
GET /resources/{resource_id}
```

Retrieve a specific owned resource

3.1.3.4.1. Description

This operation retrieves information about an owned resource.

3.1.3.4.2. Parameters

Path Parameters

Name	Description	Required
resource_id	Unique Resource ID	X

Header Parameters

Name	Description	Required
Authorization	JWT or Bearer Token	-

3.1.3.4.3. Return Type

[resource]

3.1.3.4.4. Content Type

- application/json

3.1.3.4.5. Responses

Table 17. http response codes

Code	Message	Datatype
200	OK	[resource]
404	NOT FOUND	<<>>

3.1.3.4.6. Samples

3.1.3.5. Resource HEAD (ID)

`HEAD /resources/{resource_id}`

Retrieve a specific owned resource

3.1.3.5.1. Description

This operation retrieves information about an owned resource.

3.1.3.5.2. Parameters

Path Parameters

Name	Description	Required
resource_id	Unique Resource ID	X

Header Parameters

Name	Description	Required
Authorization	JWT or Bearer Token	-

3.1.3.5.3. Return Type

[resource]

3.1.3.5.4. Content Type

- application/json

3.1.3.5.5. Responses

Table 18. http response codes

Code	Message	Datatype
200	OK	[resource]
404	NOT FOUND	<<>>

3.1.3.5.6. Samples

3.1.3.6. Resource PUT (ID)

`PUT /resources/{resource_id}`

Updates an existing Resource reference in the Platform

3.1.3.6.1. Description

This operation updates an existing 'owned' resource reference.

3.1.3.6.2. Parameters

Path Parameters

Name	Description	Required
resource_id	Unique Resource ID	X

3.1.3.6.3. Body Parameter

Name	Description	Required
Resource	*Resource*	X

Header Parameters

Name	Description	Required
Authorization	JWT or Bearer Token	-

3.1.3.6.4. Return Type

-

3.1.3.6.5. Responses

Table 19. http response codes

Code	Message	Datatype
200	OK	<<>>
401	UNAUTHORIZED	<<>>
404	NOT FOUND	<<>>

3.1.3.6.6. Samples

3.1.3.7. Resource PATCH (ID)

`PATCH /resources/{resource_id}`

Patches an existing Resource reference in the Platform
Currently, only support for this method is presented, and requires and functions the same as PUT.

3.1.3.7.1. Description

This operation updates an existing 'owned' resource reference.

3.1.3.7.2. Parameters

Path Parameters

Name	Description	Required
resource_id	Unique Resource ID	X

3.1.3.7.3. Body Parameter

Name	Description	Required
Resource	*Resource*	X

Header Parameters

Name	Description	Required
Authorization	JWT or Bearer Token	-

3.1.3.7.4. Return Type

-

3.1.3.7.5. Responses

Table 20. http response codes

Code	Message	Datatype
200	OK	<<>>
401	UNAUTHORIZED	<<>>
404	NOT FOUND	<<>>

3.1.3.7.6. Samples

3.1.4. API

3.1.4.1. Swagger UI

`/swagger-ui`

3.1.4.1.1. Description

This operation accesses the API for the Policy Enforcement Point

3.1.4.1.2. Parameters

Path Parameters

Name	Description	Required
-	-	-

Header Parameters

Name	Description	Required
-	-	-

3.1.4.1.3. Return Type

-

3.1.4.1.4. Responses

Table 21. http response codes

Code	Message	Datatype
200	OK	<<>>

3.1.4.1.5. Samples

3.2. Models

3.2.1. *NewResource*

Field Name	Required	Type	Description	Format
name	Y	String	Human readable name for the resource	-
description	Y	String	Human readable description of the resource	-

Field Name	Required	Type	Description	Format
icon_uri	Y	String	Protected uri of the resource.	-
resource_scopes	Y	List of [string]	List of scopes associated with the resource	-

3.2.2. *Resource*

Field Name	Required	Type	Description	Format
ownership_id	Y	UUID	UUID of the Owner End-User	uuid
description	Y	UUID	Human readable description of the resource	uuid
name	Y	String	Human readable name for the resource	-
icon_uri	Y	String	Protected uri of the resource.	-
resource_scopes	Y	List of [string]	List of scopes associated with the resource	-

<< End of Document >>