FIWARE Authorization PDP API Specification

DATE: 16 September 2016

This version:
http://authzforce.github.io/fiware/authorization-pdp-api-spec/5.2

Previous version:
http://authzforce.github.io/fiware/authorization-pdp-api-spec/4.4

Latest version:
http://authzforce.github.io/fiware/authorization-pdp-api-spec/latest

Editors

- Cyril DANGERVILLE, Thales Services

Copyright

Abstract

This specification defines a RESTful API of an Authorization Policy Decision Point (PDP) compliant with the OASIS XACML standard. More specifically, it defines RESTful interfaces for:

- Managing XACML-compliant authorization policies;
- Requesting authorization decisions based on those policies, in a XACML-compliant request-response format;
- Managing multiple PDPs, one per domain (aka tenant), in order to provide multi-tenancy.

This APIARY blueprint gives a user-friendly description of the API. However, the official API reference is available in a WADL (Web Application Description Language) and XML schema files on the Github repository of AuthZForce REST API model project. From this WADL (and associated XSD files), you can generate code automatically for various languages, e.g. Apache CXF's WADL2Java plugin for Java.

Status of this document

This is a work in progress and is changing on a daily basis.

Table of Contents

API Summary 3
Terminology 3
Policy and PolicySet 3
PDP (Policy Decision Point) 3
PAP (Policy Administration Point) 4
Domain 4
License 4
Conformance 4
API Specification 5
Domains 5
Resource for all Domains 5
Add domain 5
Get domains 5
Domain 5
Resource of a specific administration Domain 6
Get Domain Sub-resources 6
Delete domain 6
Resource Domain properties 6
Update domain properties 6
Get domain properties 6
Domain Policy Administration Point 6
Resource Domain PAP - Policies 7
Add/update a policy 7
Get policies 7
Resource Domain PAP - Policy 7
Get policy 7
Delete policy 7
Resource Domain PAP - Policy version 8
Get policy version 8
Delete policy version 8
Resource Domain PAP - PDP properties 9
Update PDP properties 9
Get PDP properties 9
Domain Policy Decision Point 9
Resource Domain PDP 9
Request authorization decision 9
Examples 9
Domains 10
Resource for all Domains 10
Add domain 10
Domain 10
Resource Domain properties 10
Update domain properties 11
Domain Policy Administration Point 11
Resource Domain PAP - Policies 11
Add/update a policy 11
Resource Domain PAP - PDP properties 12
Update PDP properties 12
Domain Policy Decision Point 13
Resource Domain PDP 13
Request authorization decision 13
References 15

API Summary

- Domains
  - Resource for all Domains
    - POST - Add domain [/domains]
    - GET - Get domains [/domains{?externalId}]
- Domain
  - Resource of a specific administration Domain
    - GET - Get Domain Sub-resources [/domains/{domainId}]
    - DELETE - Delete domain [/domains/{domainId}]
  - Resource Domain properties
    - PUT - Update domain properties [/domains/{domainId}/properties]
    - GET - Get domain properties [/domains/{domainId}/properties]
- Domain Policy Administration Point
  - Resource Domain PAP - Policies
    - POST - Add/update a policy [/domains/{domainId}/pap/policies]
    - GET - Get policies [/domains/{domainId}/pap/policies]
  - Resource Domain PAP - Policy
    - GET - Get policy [/domains/{domainId}/pap/policies/{policyId}]
    - DELETE - Delete policy [/domains/{domainId}/pap/policies/{policyId}]
  - Resource Domain PAP - Policy version
    - GET - Get policy version [/domains/{domainId}/pap/policies/{policyId}/{version}]
    - DELETE - Delete policy version [/domains/{domainId}/pap/policies/{policyId}/{version}]
  - Resource Domain PAP - PDP properties
    - PUT - Update PDP properties [/domains/{domainId}/pap/pdp.properties]
    - GET - Get PDP properties [/domains/{domainId}/pap/pdp.properties]
- Domain Policy Decision Point
  - Resource Domain PDP
    - POST - Request authorization decision [/domains/{domainId}/pdp]

Terminology

Policy and PolicySet

A *Policy* is a set of *Rules*, and a *PolicySet* is a set of *Policy* elements. A *Rule* consists of a condition on the access request attributes, and a decision – *Permit* or *Deny* - to apply if the condition holds true for the request. A *Policy* (resp. *PolicySet*) combines multiple *Rules* (resp. *Policies*) and therefore multiple decisions together in various ways (defined in the standard) to make the final decision.

PDP (Policy Decision Point)

The PDP provides authorization decisions based on various attributes given at runtime by PEPs (Policy Enforcement Points) about each incoming access request, and XACML policies that define multiple rules checking whether those attributes (and therefore the access request) satisfy certain conditions. The attributes provided by the PEP (see below) about each access request may be attributes about the request itself: The request URL, the HTTP method; about the requester: The access requester ID, requester role. The PDP may add attributes to the context on its own, such as the current date and time when the requested is received. By replacing all the attribute references in the policy with these input values, PDP is able to evaluate the policy and determine whether the access should be granted.

PAP (Policy Administration Point)

The PAP provides an interface for policy administrators to manage XACML policies to be enforced by the PDP. This endpoint is provided by the Authorization PDP GE as well as a RESTful API interface. The IdM GE also provides a form of graphical interface for the PAP, as part of its access management feature. This feature actually uses the Authorization PDP GE's PAP API as backend.

Domain

The API is designed to be multi-tenant, i.e. it allows users or organizations to work on authorization policies in complete isolation from each other. In this document, we use the term *domain* instead of *tenant*. In this context, a domain mostly consists of a specific dedicated PDP with specific policies. We may use the terms *domain*, *administration domain* and *policy administration domain* interchangeably in this document.

License

This specification is licensed under the FIWARE Open Specification License (implicit patents license).

Conformance

All the interfaces described by this specification are mandatory and must be implemented in order to be compliant with.

API Specification

Domains

Resource for all Domains **[/domains]**

Policy administration domains.

Add domain

POST /domains

Create a new administration domain with defined properties, including one called `externalId`. We use the same definition for this property as in the SCIM schema, § 3.1. The response is the relative link to the REST resource created for the domain with a unique (opaque) ID assigned by the service.

Get domains

GET /domains{?externalId}

Retrieve links to domains.

Parameters

externalId (optional, string)
If specified, only a link to the domain with a matching `externalId` is returned, or none if no match.

Domain

Resource of a specific administration Domain

[/domains/{domainId}]

Policy administration domain, dedicated to specific tenant or project or workspace, logically isolated from other domains

Parameters

`domainId` (required, string)
Domain ID

Get Domain Sub-resources

GET /domains/{domainId}

Retrieves links to sub-resources in the domain. This must include a link to the PDP (`/pdp` as described later on) as specified by the REST Profile of XACML v3.0 Version 1.0, in test assertion urn:oasis:names:tc:xacml:3.0:profile:rest:assertion:home:pdp.

Delete domain

DELETE /domains/{domainId}

Delete the domain.

Resource Domain properties **[/domains/{domainId}/properties]**

Policy administration domain properties

Update domain properties

PUT /domains/{domainId}/properties

Update the properties of the domain. In this example, we change the `externalId` and description.

Get domain properties

GET /domains/{domainId}/properties

Get the properties of the domain.

Domain Policy Administration Point

Resource Domain PAP - Policies

[/domains/{domainId}/pap/policies]

Policy Administration Point interface to manage the policies of the domain.

Add/update a policy

POST /domains/{domainId}/pap/policies

Add/update a policy (XACML `PolicySet`) in the domain. The response is the relative link to the REST resource created for the policy version. If there is no existing policy with a `PolicySetId` matching the one in the request, this adds a new policy resource. If a policy exists with same `PolicySetId` but a different `Version`, this adds a new resource for this new policy version under the policy resource. If a policy exists with the same `PolicySetId` and `Version`, this must raise a conflict error and the operation must be canceled.

Get policies

GET /domains/{domainId}/pap/policies

Get the list of domain's policies. The response is the list of links to all policy resources.

Resource Domain PAP - Policy

[/domains/{domainId}/pap/policies/{policyId}]

A policy in the domain.

Parameters

policyId (required, string)
Policy ID, more precisely the XACML `PolicySetId`.

Get policy

GET /domains/{domainId}/pap/policies/{policyId}

Get the policy, i.e. the links to all versions of the policy.

Delete policy

DELETE /domains/{domainId}/pap/policies/{policyId}

Delete the policy, i.e. all policy versions. The response is the list of all the versions of the removed policy before removal.

Resource Domain PAP - Policy version

[/domains/{domainId}/pap/policies/{policyId}/{version}]

A policy version.

Parameters

version (required, string)
Policy version, more precisely the XACML `PolicySet` `Version`.

Get policy version

GET /domains/{domainId}/pap/policies/{policyId}/{version}

Get a specific version of a policy.

Delete policy version

DELETE /domains/{domainId}/pap/policies/{policyId}/{version}

Delete a specific version of a policy. The response is the removed policy version.

Resource Domain PAP - PDP properties

[/domains/{domainId}/pap/pdp.properties]

Policy Administration Point interface to manage PDP properties:

- One or more optional `feature` elements: extra implementation-specific feature ID. A `feature` element may have a `type` attribute to indicate the category/type of feature, and a `enabled` attribute to indicate whether the feature is actually enabled or not.
- One mandatory `rootPolicyRefExpression` element of type `{urn:oasis:names:tc:xacml:3.0:core:schema:wd-17}IdReferenceType` (XACML v3.0): expression (with version pattern(s)) to be matched by the PolicySet's `PolicySetId` and `Version` to be enforced by the PDP as root policy. This policy and any other policy referenced (directly or indirectly) from it via XACML PolicySetIdReference must correspond to a resource under URL path '/policies'. The documentation of any implementation of this GE must mention what is the default value for this element.

Update PDP properties

PUT /domains/{domainId}/pap/pdp.properties

Get PDP properties

GET /domains/{domainId}/pap/pdp.properties

Get a specific version of a policy.

Domain Policy Decision Point

Resource Domain PDP **[/domains/{domainId}/pdp]**

Policy Decision Point.

Request authorization decision

POST /domains/{domainId}/pdp

Examples

Domains

Resource for all Domains

[/domains]

Add domain

POST /domains

Request /domains (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<domainProperties xmlns="http://authzforce.github.io/rest-api-model/xmlns/authz/5"
    externalId="my.test.domain">
    <description>This is a test domain.</description>
</domainProperties>
```

Response 200 (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<link xmlns="http://www.w3.org/2005/Atom" rel="item" href="1234ABCD" title="1234ABCD"/>
```

Domain

Resource Domain properties

[/domains/{domainId}/properties]

Update domain properties

PUT /domains/{domainId}/properties

Request /domains/{domainId}/properties (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<domainProperties xmlns="http://authzforce.github.io/rest-api-model/xmlns/authz/5" 
    externalId="my.test.domain0">
    <description>This is a test domain and I want to use it.</description>
</domainProperties>
```

Response 200 (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<domainProperties xmlns="http://authzforce.github.io/rest-api-model/xmlns/authz/5" 
    externalId="my.test.domain0">
    <description>This is a test domain and I want to use it.</description>
</domainProperties>
```

Domain Policy Administration Point

Resource Domain PAP - Policies

[/domains/{domainId}/pap/policies]

Add/update a policy

POST /domains/{domainId}/pap/policies

Request /domains/{domainId}/pap/policies (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17" PolicySetId="P1" Version="1.0" ... />
```

Response 200 (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<link xmlns="http://www.w3.org/2005/Atom" rel="item" href="P1/1.0" title="Policy 'P1' v1.0"/>
```

Resource Domain PAP - PDP properties

[/domains/{domainId}/pap/pdp.properties]

Update PDP properties

PUT /domains/{domainId}/pap/pdp.properties

Request /domains/{domainId}/pap/pdp.properties (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<pdpPropertiesUpdate
    xmlns="http://authzforce.github.io/rest-api-model/xmlns/authz/5">
    <feature
             type="urn:ow2:authzforce:feature-type:pdp:request-filter"
        enabled="true">urn:ow2:authzforce:feature:pdp:request-filter:multiple:repeated-attribute-categories-lax</feature>
    ...(content omitted)...
    <rootPolicyRefExpression>root</rootPolicyRefExpression>
```

```
</pdpPropertiesUpdate>
```

Response 200 (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<pdpProperties
    xmlns="http://authzforce.github.io/rest-api-model/xmlns/authz/5"
      lastModifiedTime="2016-05-28T14:21:35.730Z">
    <feature
            type="urn:ow2:authzforce:feature-type:pdp:request-filter"
        enabled="false">urn:ow2:authzforce:feature:pdp:request-filter:multiple:repeated-attribute-categories-strict</feature>
    <feature
        type="urn:ow2:authzforce:feature-type:pdp:request-filter"
        enabled="true">urn:ow2:authzforce:feature:pdp:request-filter:multiple:repeated-attribute-categories-lax</feature>
    ...(content omitted)...
    <rootPolicyRefExpression>root</rootPolicyRefExpression>
</pdpProperties>
```

Domain Policy Decision Point

Resource Domain PDP

[/domains/{domainId}/pdp]

Request authorization decision

POST /domains/{domainId}/pdp

Request /domains/{domainId}/pdp (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<Request xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17" ... />
```

Response 200 (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<Response xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17" ... />
```

References

- Apiary project (http://docs.authorizationpdp.apiary.io/#reference)
- Github source (https://github.com/authzforce/fiware.git)
- Github repository of AuthZForce REST API model project (https://github.com/authzforce/rest-api-model/tree/release-5.2.0/src/main/resources)
- THALES (http://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Thales_sv)
- FIWARE Open Specification License (implicit patents license) (https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FI-WARE_Open_Specification_Legal_Notice_%28implicit_patents_license%29)
- SCIM schema, § 3.1 (https://tools.ietf.org/html/rfc7643#section-3.1)
- REST Profile of XACML v3.0 Version 1.0 (http://docs.oasis-open.org/xacml/xacml-rest/v1.0/xacml-rest-v1.0.html)
- urn:oasis:names:tc:xacml:3.0:profile:rest:assertion:home:pdp (http://docs.oasis-open.org/xacml/xacml-rest/v1.0/cs02/xacml-rest-v1.0-cs02.html#_Toc399235433)