**Mulesoft Policies**

*by* ***Fernando Ramires*** *on* ***01 December of 2022***

1. **What are policies?**

*“Policies enable you to enforce regulations to help manage security, control traffic, and improve adaptability of your APIs. For example, a policy can control authentication, access, allotted consumption, and service level access (SLA).”*

Enforce these regulations to APIs without any code changes. Policies could be applied across any APIs implemented on Anypoint Platform as well as using proxies for APIs implemented out of Anypoint Platform.

API Manager provides default policies providing handling different aspects of security, management and governance. **In addition to the default policies, custom policies could also be created and applied to suit any custom needs.**

1. **Default Policy Categories**

*Default Policies available within API Manager are mainly categorized based on functionality they provide. Categories available are:*

**Security**: Policies available in this category are for enforcing Authentication (Choose between methods such as Basic, OAuth2.0, JWT, etc.), allow / deny access based on IP Range (IP Black / Whitelisting).

**Compliance**: Enforce requirements such as Client ID and CORS.

**Transformation**: Allows to transform / enrich headers for request or response. This could be used to add or remove any default or sensitive information from within the headers.

**Quality of Service:** QoS based policies mainly provide SLA support, Spike control to queue traffic and Caching.

**Troubleshooting**: For troubleshooting incoming requests by logging messages.

1. **Policies applicated in Experience APIs**

*Since these APIs are more front line and often exposed out, policies that enforce protection from threats, API security and usage are essential.*

*Policies Applicable: OAuth, JWT, JSON / XML Threat Protection, Rate Limiting, SLA Based, Caching*

**Authentication:** Use one of OAuth 2.0 or JWT policies to protect from anonymous usage. If only LDAP is available, then use Basic Authentication — LDAP. Additional Auth

**Protection:** Use JSON / XML threat protection policies to protect against dangerous payloads. Additionally, IP Blacklist policy could be used to block traffic from frequent violators or well-known dark nets.

**Access:** When API is consumed by a Web Application, use CORS to enable access for enable or disable access from other domains.

**Usage Monitoring & Limits:** Use Rate limiting / SLA-Based policy to enforce limitations on API usage.

**Performance:** Use HTTP caching to cache responses and optimize resource usage by serving cached data. Caching could be limited based on time, method or further customized based on need when applying.

**Customize Headers**: Use Header Injection to add custom meta data for processing requests downstream or for responses such as to security signatures etc. Similarly use Header removal to remove any sensitive data being sent through headers such as HSTS info.

1. **Policies applicated in Process APIs**

*These APIs are central and essentially handling core business rules and coordinating requests to one or more downstream system APIs. Policies that help with performance yet enforce some security without having impact on performance is a common requirement.*

*Policies Applicable: Basic Authentication, Spike Control, IP Whitelisting, Rate Limiting, Caching, Header Injection & Removal.*

**Authentication:** Since Process APIs are usually internal and mostly consumed by Experience APIs, Basic Authentication would be ideal.

**Protection:** In some cases, spike control could help to limit amount of requests being sent to downstream APIs if downstream APIs are under performing due an outage / when using DR Services.

**Access:** IP Whitelisting could be used to limit access to a dedicated set of consumers. This policy accepts IP(s) and IP Range(s). Use this policy to whitelist IP’s of upstream APIs (Experience APIs).

**Usage Monitoring & Limits:** Use Rate Limiting policies to track and limit usage.

**Performance:** Use HTTP caching to cache responses and optimize resource usage by serving cached data. Caching could be limited based on time, method or further customized based on need when applying.

**Customize Headers:** Remove headers returned from downstream APIs using Header Removal Policy or Inject headers such as Client IDs using Header Injection Policy.

1. **Policies applicated in System APIs**

*These APIs are highly used and often communicating with data sources or third-party systems. Reliability and performance would be key elements to factor in.*

*Policies Applicable: Basic Authentication, Spike Control, IP Whitelisting, Rate Limiting, Caching, Header Injection & Removal.*

**Authentication:** Like Process APIs, System APIs are internal and mostly consumed by Process APIs, Basic Authentication would be ideal policy for Authentication.

**Protection:** Use spike control if System APIs provide access to legacy systems that cannot multitask or accept huge amount of requests.

**Access:** IP Whitelisting could be used to limit access to a dedicated set of consumers. This policy accepts IP(s) and IP Range(s). Whitelist upstream API IP Addresses (process APIs).

**Usage Monitoring & Limits:** Use Rate Limiting policies to limit usage of API or a specific operation(s). For ex: a third-party app only allows certain amount of operations per day. Use this to limit # of transactions being submitted.

**Performance:** Use HTTP caching to cache responses and optimize resource usage by serving cached data. Caching could be limited based on time, method or further customized based on need when applying.

**Customize Headers**: Remove headers returned from downstream APIs / Systems using Header Removal Policy.

1. **Disadvantages of uses of policies**
2. Increased number of policies could affect APIs performance as they are executed in the runtime.
3. If poorly implemented, it can be confusing in evolution of development/support for APIs.
4. The use of custom policy requires a higher level of knowledge of the team in the Anypoint plataform.
5. **References**

* [Policies Overview | MuleSoft Documentation](https://docs.mulesoft.com/policies/policies-policy-overview)
* [Policy Types | MuleSoft Documentation](https://docs.mulesoft.com/policies/policies-policy-types)
* [Policy Categories | MuleSoft Documentation](https://docs.mulesoft.com/policies/policies-policy-categories)
* [Anypoint API Manager | MuleSoft Documentation](https://docs.mulesoft.com/api-manager/2.x/latest-overview-concept)
* [Gateway Policies | MuleSoft Documentation](https://docs.mulesoft.com/policies/policies-overview)
* [Applying Automated Policies | MuleSoft Documentation](https://docs.mulesoft.com/policies/policies-automated-applying)
* [MuleSoft Policies: Govern and Manage APIs efficiently | Medium](https://medium.com/@pkurimella/mulesoft-policies-govern-and-manage-apis-efficiently-5dee0d7656db)