**Topic:**

**Difference between sync and async?**

Sync or Synchronous follows a request-response strategy, that is, it stops all its processing until it gets a response from the external source.

Async or asynchronous follows a one-way strategy which does not wait for the response.

Sync runs on a single thread whereas async runs on multi-thread.

Sync enables the sequential thread processing whereas async allows requests or subflows to process in parallel to the calling flow.

**How can we achieve async processing in mule?**

Ref: <https://docs.mulesoft.com/mule-runtime/4.3/async-scope-reference>

Async processing can be achieved by using an “async” scope in mule applications.

To facilitate async scope, the main flow passes the message to async scope to trigger it and parallely sends a copy of the message to the next processor in the flow after the async scope.

Async scope does not inherit the exception strategy of the main flow.

**When do you want to choose async?**

Async is chosen to process time-consuming operations that do not need us to send a response back, like printing, connecting to a mail server, writing to a file.

**Use of target variable?**

Target variables are used to store the contents of a response message so that the original payload is not lost.

**Example: REST API with XML Payload**

POC in Anypoint Studio

**Connector: Workday Connector**

Workday, Inc. is an American on‑demand financial management and human capital management software vendor

**Purpose**

The Workday talent management suite includes tools to improve managing, developing, and rewarding employees. It contains functions perfect for management, including onboarding, goal management, performance management, succession planning, and career and development planning.

Ref: <https://anypoint.mulesoft.com/exchange/com.mulesoft.connectors/mule-workday-connector/>

The **Workday connector** allows businesses to extend the capabilities of Workday and enable seamless interoperability with third-party SaaS applications, systems, and services that are vital to the human resources ecosystem, such as recruiting, talent management, core HRM, payroll, benefits, and more. Create connectivity to essential applications such as SAP, PeopleSoft, Taleo, ADP, NetSuite and Salesforce through numerous Workday APIs, such as the Workday Payroll, Time Tracking, Procurement, and Expense Management.

**Security Policy: HTTP Caching**

Ref: <https://docs.mulesoft.com/api-manager/2.x/http-caching-policy>

This policy provides a way to store HTTP responses from an API implementation or an API proxy for later reuse. This policy avoids performing multiple calls to the backend when the response of a service does not change often and to optimize against computationally expensive processing. This policy uses the concept of a cache which stores data so future requests for that data can be served faster

This policy stores the whole HTTP response as received by the policy including the headers, payload, status code and session variables. The policy also stores any changes made by any other policy before reaching the caching policy but can not store the changes made by any other policy after passing through caching policy.

Each entry has a size limit of 1MB. If the content-length information is not provided in the HTTP response, the policy stores data till the maximum limit and then continues without storing, after limit has exceeded.

How it works?

When the policy receives a request, it calculates the key under which the response would be saved. When a second request with the same key arrives, the policy checks the cache and returns the response without sending the request to following policies, API implementation or API proxy. By default, the HTTP Caching key is the requestPath of the endpoint. (If you call the endpoint as http://myAppUrl.com/my/policy the key is /my/policy.)

The store used as cache is Mule’s ObjectStore. Based on the policy configuration and where it is deployed, the policy uses an in-memory, persistent, file-based object store or the Runtime Manager ObjectStore v2.

Policy configuration topics:

* [HTTP Caching Key](https://docs.mulesoft.com/api-manager/2.x/http-caching-policy#http-caching-key)
* [Maximum Cache Entries](https://docs.mulesoft.com/api-manager/2.x/http-caching-policy#maximum-cache-entries)
* [Entry Time To Live](https://docs.mulesoft.com/api-manager/2.x/http-caching-policy#entry-time-to-live)
* [Distributed](https://docs.mulesoft.com/api-manager/2.x/http-caching-policy#distributed)
* [Persist Cache](https://docs.mulesoft.com/api-manager/2.x/http-caching-policy#persist-cache)
* [Follow HTTP Caching Directives](https://docs.mulesoft.com/api-manager/2.x/http-caching-policy#follow-http-caching-directives)
* [Invalidation Header](https://docs.mulesoft.com/api-manager/2.x/http-caching-policy#invalidation-header)
* [Conditional Request Caching Expression](https://docs.mulesoft.com/api-manager/2.x/http-caching-policy#conditional-request-caching-expression)
* [Conditional Response Caching Expression](https://docs.mulesoft.com/api-manager/2.x/http-caching-policy#conditional-response-caching-expression)

POC in anypointStudio (Run command in cmd: curl -vvv X GET http://caching-policy-demo-sls.us-e2.cloudhub.io/flights?destination=LAX)