

Token Utility using DistributedMapCache

This template fetches a token from a given URL and stores it in the DistributedMapCache. The token can then be fetched from the cache whenever required (if the token is in its TTL) instead of making the API call every time.

Key Components :

1. GenerateFlowfile – Generates flowfile with the username and password required to make the API call for the access token.
2. InvokeHTTP – Used for the API call to get the access token.
3. EvaluateJsonPath – Extracts the access token into the flowfile content.
4. PutDistributedMapCache – Stores the incoming flowfile's content in the cache with the variable name as defined in the '*Cache Entry Identifier*' property of the processor.
5. FetchDistributedMapCache – Retrieves the value of the access token from the cache and puts it in an attribute which takes the name as defined in the '*Put Cache Value In Attribute*' property of the processor.

Controller Services Used :

1. DistributedMapCacheClientService – Provides the ability to communicate with a DistributedMapCacheServer. This can be used to share a Map between nodes in a NiFi cluster.
2. DistributedMapCacheServer – Provides a map (key/value) cache that can be accessed over a socket. Interaction with this service is typically accomplished via a DistributedMapCacheClient service.

NOTE :

1. The LogAttribute processors signify a logical termination. If further processing is required, other processors may be used in their place.