

Psyncopate Replay Service for MuleSoft

OVERVIEW

Psyncopate Replay Service for MuleSoft (Replay Service) simplifies error recovery by automating the process. Clients can pass the payload and associated meta-data of a failing event, along with the amount of time to wait before trying again, to the Replay Service, which in turn will mediate recovery without further intervention.

Failures in software are a fact of life. Required services can be down, databases can be offline, or user error can and does happen. When failures are encountered, a typical requirement is that the software make some attempt to recover, usually by retrying the failed operation. Platforms like MuleSoft provide intrinsic retry capability such as reconnection strategies and retry scopes. However, some critical failures cannot be recovered using native mechanisms without significant additional development effort. For example, if the recovery requirements mandate that the time between retries must be measured in minutes, hours or days; or require manual intervention, then the impact on resources and performance render native retry mechanisms unsuitable. For uses-cases where these native approaches to error recovery will not work, the Replay Service is a preferred solution.

FEATURES

- **FAILURE RECOVERY**
Supports recovery of failed operations either automatically after a period of time elapses or on demand.
- **RESTFUL INTERFACE**
The service provides a full-featured ReST interface through which client requests are serviced.
- **CONFIGURABLE RETRY**
Should a recovery attempt by the replay service fail, the attempt can be automatically retried using client specified retry limits and time between attempts.

- **EXTENSIBLE**
The architecture and design of the Replay Service inherently supports application specific customization and extension if required.

BENEFITS

- **MANUAL RECOVERY**
Failures are often caused by bad data or coding errors. This kind of error cannot be automatically recovered. The Replay Services provides support for a person to correct the failing data or correct the coding error and replay the failed operation.
- **AUTOMATIC RECOVERY**
When a client determines that an error is recoverable if retried after a long wait, the Replay Service allows the client to store a replay message and automatically retry it at some specified time in the future. The client may also set other important recovery parameters when requesting replay services.
- **FASTER TIME TO MARKET**
The Replay Services provides an out-of-the-box solution for error recovery that significantly reduces the time and money that would otherwise be required to implement complex and time-consuming custom retry and error recovery capabilities.
- **FLEXIBILITY**
The Replay Service supports ReST callbacks and can support message-based (Anypoint MQ or JMS), email and File Transfer Service (FTP) replay mechanisms – all out of the box.

REQUIRED SOFTWARE

- Microsoft SQL Server 2017 or higher; or,
- MySQL 8.x or better
- MuleSoft Runtime 4.3.0 or higher

Psyncope Replay Service for MuleSoft

DESCRIPTION

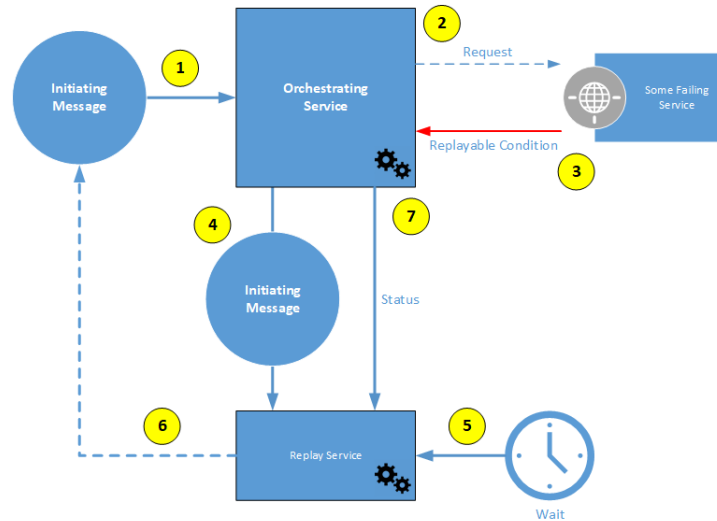
The domain of problems that the replay service addresses is shown in the figure below. Computing actions begin with an initiating message, which is denoted by the yellow circle labeled '1'. An initiating message could be a web service call, a message delivered from a message service such as JMS or Anypoint MQ, or some other external action. Initiating messages are received and processed by an orchestrating service. The orchestrating service may perform one or more actions on the initiating message such as a subsequent request to another service as shown by the yellow circle labeled '2'. This second service is not necessarily a web service, although it could be. The action labeled "Some Failing Service" could be a request to an external database, a file read or write, or an internal transformation the orchestrating service attempted and that failed.

Some of these failing requests can be retried using native mechanisms; however, others, as shown in the figure labeled by the yellow circle labeled '3' are more rightly "replay-able" rather than "retry-able." A "replay-able" condition is one that may correct itself after a certain amount of time elapses. The key difference between a "retry-able" condition and a "replay-able" condition is the amount of time that must elapse between the time the error condition occurred and when the action can be reattempted. Usually a *retry-able* condition is one that can be reattempted within a few milliseconds or a few seconds.

There are two kinds of *replay-able* conditions. One kind can be recovered automatically by simply retrying it automatically at some future time, either minutes or hours in the future. This is referred to as an "automated"

replay. A second kind of "replay-able" condition, referred to as a "journal" or "manual" replay, is one that may succeed if some manual intervention occurs, such as a credential change, or even a change to the payload of the initiating message itself.

When any *replay-able* condition occurs, if the orchestrating service sends the message to the replay service as shown by the circle labelled '6', then the replay service can mediate automatic recovery. After time elapses (5), the replay service posts the initiating message back to the orchestrating service (6), and the status is updated (7), closing the loop.



Status could mean that the replay action succeeded or that it failed. A failure may be recoverable and thus retried again after waiting a period of time (which the replay service will do automatically). A failure may also be unrecoverable. Unrecoverable failures cannot be recovered using automation; however, the Replay Service also helps when this occurs by storing

the original failing message and its metadata for forensic analysis, or manual correction and recovery using baked in capabilities.

For more information, e-mail: info@psyncope.com