Table of Contents

[Talend API Introduction 2](#_Toc189050976)

[Orchestration 2](#_Toc189050977)

[Connection API 2](#_Toc189050978)

[Overview: 2](#_Toc189050979)

[Key Components: 3](#_Toc189050980)

[API Endpoints: 3](#_Toc189050981)

[Key Features 4](#_Toc189050982)

[Tasks API Use case 5](#_Toc189050983)

[Use Case Overview: 5](#_Toc189050984)

[Artifacts 9](#_Toc189050985)

[Overview: 9](#_Toc189050986)

[Key Components: 9](#_Toc189050987)

[API Endpoints: 9](#_Toc189050988)

[Practical Example: 9](#_Toc189050989)

[Key Components and Their Use Cases: 10](#_Toc189050990)

[Environments : 13](#_Toc189050991)

[Workspaces : 14](#_Toc189050992)

[Monitoring task runs 14](#_Toc189050993)

[Use Case Overview: 14](#_Toc189050994)

[Execution Logs 16](#_Toc189050995)

[Data Model Overview 17](#_Toc189050996)

[Entities 17](#_Toc189050997)

[Relationships 18](#_Toc189050998)

[Results: 20](#_Toc189050999)

# Talend API Introduction

Talend offers a comprehensive suite of APIs designed to facilitate seamless integration, management, and monitoring of data processes within its cloud environment. Below is an overview of each API category, along with detailed explanations and practical use cases.

# Orchestration

Here we Manage all Artifacts, Tasks, Plans, Schedules, Environments, Workspaces, Promotions and all associated Ressources

# Connection API

## Overview:

The **Connection API** provides a set of endpoints to manage connections used by datasets in Talend Cloud. Connections define the configurations and credentials necessary to access external data sources like databases, file systems, or APIs. These connections can be reused across multiple datasets and tasks, making them essential for efficient data integration and management.

## Key Components:

* **UpdateConnectionRequest:** Request body for updating a connection.
* **CreateConnectionRequest:** Structure for creating a new connection.
* **ConnectionParameter:** Defines properties such as host, username, and password.
* **ConnectionDetails:** Contains full information about a connection.
* **PageConnection:** Paginated response containing multiple connections.

## API Endpoints:

1. **Create a Connection** → POST /orchestration/connections
2. **Get available Connections** → GET /orchestration/connections
3. **Update a Connection** → PUT /orchestration/connections/{connectionId}
4. **Delete a Connection** → DELETE /orchestration/connections/{connectionId}
5. **Get Connection details** → GET /orchestration/connections/{connectionId}

**Reference:** [Talend Connections API](https://api.talend.com/apis/orchestration/2021-03/)

Ref: [Talend Management Console :: Public API :: Documentation](https://api.eu.cloud.talend.com/tmc/swagger/swagger-ui.html#/)

**Request URL**

<https://api.eu.cloud.talend.com/tmc/v2.6/connections>

Response :

A screenshot of a computer program

Description automatically generated

Mysql Connection table :

Ref: [TALEND\_COMPONENT\_KIT/connection.py at main · VenkMohanDate/TALEND\_COMPONENT\_KIT](https://github.com/VenkMohanDate/TALEND_COMPONENT_KIT/blob/main/connection.py)

A screenshot of a computer

Description automatically generated

## Key Features

1. **Create New Connections:**
   * Add new connections to Talend Cloud for accessing external data sources.
2. **Update Existing Connections:**
   * Modify connection details, such as credentials or configurations, without affecting dependent datasets or tasks.
3. **Retrieve Connections:**
   * Fetch the list of available connections or details about a specific connection.
4. **Delete Connections:**
   * Remove unused connections to maintain a clean and organized environment.
5. Monitoring and Governance :
   * **Scenario:** A data governance team needs to audit the connections used in Talend Cloud.
   *  **Use Case:** Use the Connection API to list all available connections and their associated metadata, such as last updated timestamp and owner.
   *  **Benefit:** Gain visibility into connections and ensure they comply with organizational policies

# Tasks API Use case

Monitoring specific tasks in Talend is essential for analysis. Talend's API facilitates this by providing Task details directly

## Use Case Overview:

The primary objective is to obtain run metrics for a specific task without generating or downloading its logs. This approach streamlines monitoring by focusing on key performance indicators, reducing the overhead associated with log management.

**Prerequisites:**

1. **Access Token Generation:**
   * For users, generate a personal access token.
   * For service accounts, generate a service account token. Note that service account tokens expire after 30 minutes; if expired, regenerate using the POST method at https://api.{env}.cloud.talend.com/security/oauth/token.

A screenshot of a phone

Description automatically generated

1. **Permissions:**
   * Ensure the user or service account has the Viewer permission for the workspace containing the task to be monitored.
2. **Retrieve Task Executions:**
   * Issue a GET request to https://api.<env>.cloud.talend.com/processing/executables/tasks/{taskId}/executions with appropriate headers and query parameters (e.g., lastDays=1, status=execution\_failed) to obtain recent executions of the specified task.

GET https://api.<env>.cloud.talend.com/processing/executables/tasks/{taskId}/executions

Headers:

Content-Type: application/json

Authorization: Bearer <your\_personal\_access\_token\_or\_service\_account\_token>

A screenshot of a computer

Description automatically generated

**Request URL**

**https://api.eu.cloud.talend.com/tmc/v2.6/executables/tasks**

**Response body**

A screen shot of a computer code

Description automatically generated

MySql Tasks table

Ref: [TALEND\_COMPONENT\_KIT/tasks.py at main · VenkMohanDate/TALEND\_COMPONENT\_KIT](https://github.com/VenkMohanDate/TALEND_COMPONENT_KIT/blob/main/tasks.py)

A screenshot of a computer

Description automatically generated

# [Artifacts](https://api.eu.cloud.talend.com/tmc/swagger/swagger-ui.html#/artifacts)

## Overview:

Artifacts in Talend Orchestration represent deployable components such as jobs, tasks, or services. The API allows for artifact management, including retrieval, versioning, and deletion.

## Key Components:

* **Artifact:** Represents a deployable unit in Talend.
* **ArtifactParameter:** Defines configurable parameters for an artifact.
* **ArtifactRequest:** Payload structure for creating or updating an artifact.
* **ArtifactVersion:** Manages different versions of an artifact.
* **PageArtifact:** Paginated response containing multiple artifacts.

## API Endpoints:

1. **Get available Artifacts** → GET /orchestration/artifacts
2. **Delete Artifact by ID** → DELETE /orchestration/artifacts/{artifactId}
3. **Get Artifact by ID** → GET /orchestration/artifacts/{artifactId}
4. **Get Artifact of a specified version** → GET /orchestration/artifacts/{artifactId}/versions/{version}
5. **Delete Artifact of a specified version** → DELETE /orchestration/artifacts/{artifactId}/versions/{version}

**Reference:** [Talend Artifacts API](https://api.talend.com/apis/orchestration/2021-03/)

## Practical Example:

Consider a scenario where to automate the deployment of a data integration job from the development environment to production. Using the Orchestration API, they can:

1. **Retrieve the Artifact:** Use the GET /orchestration/artifacts endpoint to list available artifacts and identify the specific job to promote.
2. **Create a Promotion Plan:** Utilize the POST /orchestration/executables/plans endpoint to define a plan that includes the promotion steps.
3. **Schedule the Promotion:** Set up a schedule using the POST /orchestration/executables/plans/{planId}/run-config endpoint to execute the promotion at a specified time.
4. **Monitor the Execution:** Access execution details via the GET /orchestration/executables/plans/{planId}/executions endpoint to ensure the promotion was successful.

## Key Components and Their Use Cases:

1. **Artifacts Management:**
   * **Purpose:** Handle data integration components such as jobs, routes, and services.
   * **Use Cases:**
     + **Retrieving Artifacts:** Fetch a list of available artifacts to understand existing components.
     + **Managing Versions:** Access specific versions of an artifact to maintain version control.
     + **Deletion:** Remove outdated or unnecessary artifacts to keep the repository clean.

URL : **https://api.eu.cloud.talend.com/tmc/v2.6/artifacts**

Response:

A computer screen shot of a program code

Description automatically generated

A screenshot of a computer

Description automatically generated

MySql Artifact table:

Ref: [TALEND\_COMPONENT\_KIT/artifact.py at main · VenkMohanDate/TALEND\_COMPONENT\_KIT](https://github.com/VenkMohanDate/TALEND_COMPONENT_KIT/blob/main/artifact.py)

A screenshot of a computer

Description automatically generated

# Environments :

## Overview:

Environments provide a logical grouping of configurations and execution contexts for orchestration.

## Key Component:

* **EnvironmentInfo:** Describes an environment's details.

## API Endpoints:

1. **Create Environment** → POST /orchestration/environments
2. **List Environments** → GET /orchestration/environments
3. **Update Environment** → PUT /orchestration/environments/{Id}
4. **Remove Environment** → DELETE /orchestration/environments/{Id}

**Reference:** [Talend Environments API](https://api.talend.com/apis/orchestration/2021-03/)

**Request URL: https://api.eu.cloud.talend.com/orchestration/environments**

Response Body :

A screen shot of a computer program

Description automatically generated

# Workspaces :

## Overview:

Workspaces are organizational units that group related projects and resources.

## Key Components:

* **Workspace:** Represents a workspace entity.
* **WorkspaceInfo:** Contains details of a workspace.
* **CreateWorkspaceRequest:** Defines the payload for workspace creation.

## API Endpoints:

1. **Create a Workspace** → POST /orchestration/workspaces
2. **List Workspaces** → GET /orchestration/workspaces
3. **Remove a Workspace** → DELETE /orchestration/workspaces/{id}
4. **Update a Workspace** → PUT /orchestration/workspaces/{id}

**Reference:** [Talend Workspaces API](https://api.talend.com/apis/orchestration/2021-03/)

**Request URL:** [**https://api.eu.cloud.talend.com/orchestration/workspaces**](https://api.eu.cloud.talend.com/orchestration/workspaces)

**Response Body :**

A computer screen shot of a program

Description automatically generated

# Schedules API

## Overview:

Schedules automate task and plan executions using triggers**.**

## Key Components:

* ScheduleTrigger: Defines when a schedule is activated.
* Schedule: Represents a scheduled execution.
* PageSchedule: Paginated schedule list.
* CRON: Defines schedule recurrence.

## API Endpoints:

1. Get all Schedules → GET /orchestration/schedules
2. Create a Schedule → POST /orchestration/schedules
3. Update a Schedule → PUT /orchestration/schedules/{scheduleId}
4. Clone a Schedule → POST /orchestration/schedules/{scheduleId}
5. Retrieve a Schedule → GET /orchestration/schedules/{scheduleId}
6. Delete a Schedule → DELETE /orchestration/schedules/{scheduleId}
7. Manage Schedule Triggers:
   * Append a trigger → POST /orchestration/schedules/{scheduleId}/triggers
   * Remove a trigger → DELETE /orchestration/schedules/{scheduleId}/triggers
   * Update an existing trigger → PUT /orchestration/schedules/{scheduleId}/triggers
8. Simulate Next Triggered Events → POST /orchestration/schedules/{scheduleId}/events
9. Retrieve Task Schedule → GET /orchestration/executables/tasks/{taskId}/schedule
10. Manage Task-Plan Schedules:
    * Associate a Schedule to a Task → PUT /orchestration/executables/tasks/{taskId}/schedule
    * Remove Task-Schedule Association → DELETE /orchestration/executables/tasks/{taskId}/schedule
    * Associate a Schedule to a Plan → PUT /orchestration/executables/plans/{planId}/schedule
    * Retrieve Plan Schedule → GET /orchestration/executables/plans/{planId}/schedule
    * Remove Plan-Schedule Association → DELETE /orchestration/executables/plans/{planId}/schedule

Reference: [Talend Schedules API](https://api.talend.com/apis/orchestration/2021-03/)

# Monitoring task runs

Monitoring task executions is essential for maintaining the reliability and efficiency of data integration workflows. Talend's API enables users to retrieve specific logs for task runs, focusing on relevant information without the need to download entire log files. This approach streamlines the monitoring process and aids in prompt issue resolution.

## Use Case Overview:

The objective is to obtain logs for task runs within a specified period in a particular environment, filtering for specific statuses and tags. This method allows users to access pertinent logs directly, facilitating efficient monitoring and troubleshooting.

**Prerequisites:**

1. **Access Token Generation:**
   * For users, generate a personal access token.
   * For service accounts, generate a service account token. Note that service account tokens expire after 30 minutes; if expired, regenerate using the POST method at https://api.{env}.cloud.talend.com/security/oauth/token.
2. **Permissions:**
   * Ensure the user or service account has the Viewer permission for the workspace containing the task to be monitored.
3. **Environment and Workspace Identification:**
   * Obtain the environment ID through API requests or from the Talend Cloud Management Console.
   * Retrieve the workspace ID via an API request (GET https://api.<your\_environment>.cloud.talend.com/orchestration/workspaces) or through the Talend Cloud Management Console (the ID appears in the URL of the Edit workspace page).

**Procedure:**

1. **Retrieve Task Executions:**
   * Issue a POST request to https://api.<env>.cloud.talend.com/processing/executables/tasks/executions with appropriate headers and a payload specifying filters such as lastDays, status, environmentId, and tag. This request retrieves a list of task runs matching the specified criteria.
2. **Obtain Logs for Each Run:**
   * For each run ID obtained from the previous step, issue a GET request to https://api.<your\_environment>.cloud.talend.com/monitoring/executions/<runId>/log with the necessary authorization header. This request retrieves the log for the specific task run.

A screenshot of a computer

Description automatically generated

# Execution Logs

Monitoring task executions is essential for maintaining the reliability and efficiency of data integration workflows. Talend's Execution Logs API (version 2021-03) provides endpoints to retrieve detailed logs for task runs, enabling efficient monitoring and troubleshooting.

Use Case Overview:

The objective is to obtain logs for specific task executions, focusing on relevant information without the need to download entire log files. This approach streamlines the monitoring process and aids in prompt issue resolution.

Prerequisites:

1. Access Token Generation:
   * For users, generate a personal access token.
   * For service accounts, generate a service account token. Note that service account tokens expire after 30 minutes; if expired, regenerate using the POST method at https://api.{env}.cloud.talend.com/security/oauth/token.
2. Permissions:
   * Ensure the user or service account has the Viewer permission for the workspace containing the task to be monitored.
3. Task Execution Identification:
   * Obtain the execution ID of the task run you need to monitor. This ID is available on the Task execution log page, reading as Task execution ID. Alternatively, you can obtain this ID via a GET call from the /executables/tasks/{taskId}/executions endpoint.

Procedure:

1. Retrieve Task Execution Logs:
   * Issue a GET request to https://api.<your\_environment>.cloud.talend.com/monitoring/executions/{id}/logs with the necessary authorization header. This request retrieves the logs for the specific task execution identified by its ID.

A close-up of a computer screen

Description automatically generated

# Data Model Overview

## Entities

1. **Tasks**
   * Represents executable tasks in the system.
2. **Connections**
   * Contains reusable configurations for accessing external systems (e.g., databases, APIs).
3. **Artifacts**
   * Represents deployable units such as jobs, routes, or data pipelines associated with tasks.

## Relationships

* **Tasks ↔ Connections**:
  + A task may use multiple connections.
  + A connection may be reused by multiple tasks.
* **Tasks ↔ Artifacts**:
  + A task is linked to a single artifact.
  + An artifact may be referenced by multiple tasks.

A black background with a black square

Description automatically generated with medium confidence

ER Diagram :

A screenshot of a computer

Description automatically generated

# Overall process of REST API to Mysql insert

A black background with white rectangles

Description automatically generated

# Results:

Talend Job to execute table inserts for the individual API requests.

A screenshot of a computer

Description automatically generated

We here can List all the task that use specific connections

A screenshot of a computer

Description automatically generated