# Octane Integration Guide for Test Evidence Upload

This document outlines the steps and APIs involved in connecting to Octane, uploading test results, organizing them using application modules and suites, and linking them to JIRA.

## 1. Prerequisites

- JIRA ID and Octane Mapping:  
 - Each SPK in Octane is associated with a `space\_id` and `workspace\_id`.  
 - These are used to form the API URL.  
 - In QZRelease2, the `space\_id` and `workspace\_id` for a given JIRA are derived using the JIRA API via a custom field.  
  
- Test Evidence Preparation:  
 - In QZRelease2, test evidence is generated using:  
 ReleaseTrain.configure('prod')  
 b = ReleaseTrain['core.devtools'].aliases['prod.new'].stages['int'].builds[0]  
 results = generate(b, regressionJiras, False, [], True)

## 2. Authentication to Octane Client

- Credentials are managed using Sandra Password Manager and stored in:  
 /prod/core/qzrelease/octane/S-{space\_id}  
  
- To authenticate:  
 - Identify the correct `space\_id` for the JIRA.  
 - Use `initConfigWithAuthOnly()` in the `OctaneAPI` class to construct the auth config and initiate communication with Octane.  
 - This can be reused for any SPK by adapting to its JIRA and credentials path.

## 3. Application Module Hierarchy

### 3.1 Root Application Module

- Octane organizes test evidence under Application Modules.  
- In QZRelease2, we use:  
 "QzRelease-2 Octane Test Evidences"  
 as the root module for all uploaded results.  
- You can define a similar root module for other platforms like QzPlatform.  
- API:  
 getOrCreateRootApplicationModuleID()

### 3.2 Sub-Application Module

- Submodules are created under the root using the Train name.  
- API:  
 createApplicationModule()

## 4. Test Suite Structure

- Test Suites are created under sub-modules.  
- Naming format: YYYYQX (based on train version timestamp).  
- Refer:  
 \_calculateEvidencePath() # under OctaneClient  
- API:  
 createTestSuite()

## 5. Suite Runs

- Each upload creates a new suite run under the suite.  
- Naming format:  
 {TrainName}-{TrainVersion}-{StageName}-{BuildVersion}  
- Similar naming can be adopted by other platforms (e.g., stable, alpha).

## 6. Uploading Test Evidence

- To upload evidence, you need:  
 - rootApplicationModuleID  
 - testSuitePath  
 - suiteRunName  
 - results  
 - jira  
  
- Upload is done using XML format.  
- API:  
 upload() # in OctaneClient  
- The method returns a task ID, useful for tracking or reporting.

## 7. Linking JIRA to Uploaded Tests

- Retrieve the relevant runs:  
 getAutoRuns() # in OctaneClient  
  
- Link the runs to JIRA:  
 updateAutoRuns() # in OctaneClient

## 8. De-Linking JIRA from Uploaded Tests

- Octane does not support deleting uploaded test evidence.  
- However, you can remove the JIRA linkage (i.e., backlog coverage).  
- Steps:  
 1. Retrieve uploaded runs:  
 getAutoRuns()  
 2. Update with delink=True:  
 updateAutoRuns(delink=True)  
  
 Note: delink defaults to False and must be explicitly set to True.