Octane-QzRelease2 Integration - Design Level Overview

# 1. Overview

This document outlines the high-level integration flow between QzRelease2 and Octane for automated test result submission and tracking. The core functionality includes identifying workspace/space ID, uploading test results, monitoring task status, and updating backlog coverage.

# 2. High-Level Flow

- User triggers the test evidence upload from QzRelease2 by entering regression Jira.  
- The backend determines the appropriate Octane workspace and space ID based on the Jira.  
- Test results are generated and formatted as XML compatible with Octane’s expectations.  
- The XML test results are uploaded to Octane using the `test-results` endpoint in batches.  
- Task status is monitored for completion.  
- Once completed, backlog coverage is updated by linking Jira to the uploaded tests.

# 3. APIs Used and Frequencies

|  |  |  |
| --- | --- | --- |
| API Endpoint | Purpose | Frequency |
| /stories | To determine space ID and workspace ID based on Jira story | Once per Jira |
| /test-results | To upload test result batches in XML format | Multiple (based on slice size) |
| /test-results/{task\_id} | To track the status of each uploaded batch task | Polled 5 times, every 5s |
| /test\_suites | To create test suite dynamically | Once per Jira |
| /application\_modules | To fetch or create test application module | Once per upload |
| /automated\_tests | To retrieve uploaded tests to verify count | Once per upload |
| /automated\_test\_runs | To verify uploaded runs | Once per upload |
| /backlog\_coverage | To update Jira-test mapping for coverage | Once per Jira |

# 4. Task Monitoring Strategy

- Each upload returns a task ID that is checked for status up to 5 times every 5 seconds.  
- If the task remains in 'RUNNING' or 'QUEUED' status after retries, it is logged for later tracking.  
- Tasks that fail or return an 'ERROR' are retried. If they still fail, the upload is aborted for that slice.  
- Final backlog coverage is only triggered if all tasks are marked 'SUCCESS'.

# 5. Error Handling & Timeout Logic

- Task failures are logged with error details.  
- Long-running tasks (beyond 10 minutes) are marked as failures and not considered for backlog update.  
- Errors during coverage linking, test suite creation, or application module creation are logged and skipped to avoid impact.