

SoapUI Integration

- [SoapUI Integration with AIQ](#)
- [Run SoapUI as a Functional Test in AIQ](#)
- [Run SoapUI as a Performance Test in AIQ](#)
- [Run Data-Enabled SoapUI Tests in AIQ](#)

SoapUI Integration with AIQ

Recommended guidance

SoapUI integration should be treated as legacy or customer-specific unless it is actively required. For current API testing, prefer Service Suite when it can cover the same requests, assertions, data handling, and reporting.

Integration review points

- SoapUI project location, version, plugins, dependencies, and execution host.
- Endpoint configuration, authentication, certificates, proxy, and firewall rules.
- Data sources, DPL mappings, assertions, and expected artifacts.
- Pass/fail mapping between SoapUI results and AIQ execution status.

Maintenance guidance

1. Keep SoapUI assets in version control with clear ownership.
2. Do not store secrets inside project files or shared logs.
3. Run small validation tests before scheduled or high-volume execution.
4. Retire or consolidate redundant SoapUI content when Service Suite replaces it.

Last updated: October 7, 2026

Run SoapUI as a Functional Test in AIQ

Recommended guidance

Run SoapUI as a functional test in AIQ when existing SoapUI projects still provide required validation. Keep the integration controlled and consider migration to Service Suite for new API coverage.

Before execution

- Confirm the SoapUI project file, version, dependencies, endpoint, environment, and credentials.
- Use secure placeholders or approved secret storage for sensitive values.
- Validate network, proxy, certificate, and firewall access.
- Define expected assertions, return codes, and artifacts.

Review workflow

1. Run the SoapUI test independently before AIQ execution.
2. Run a small AIQ wrapper and confirm output handling.
3. Review logs and generated reports for the first meaningful failure.
4. Document legacy dependency and replacement options if applicable.

Last updated: October 7, 2026

Run SoapUI as a Performance Test in AIQ

Recommended guidance

Use SoapUI for performance testing only when the SoapUI project is thread-safe, data-ready, and still required by the customer workflow. For new API performance tests, evaluate Service Suite or native AIQ performance assets first.

Performance checks

- Thread safety, setup and teardown behavior, and shared state inside the SoapUI project.
- Data uniqueness, account pools, cleanup, and endpoint rate limits.
- Load generator CPU, memory, network, and artifact storage.
- TPS, response-time, error-rate, ramp-up, and duration targets.

Validation workflow

1. Run one thread first and verify assertions and cleanup.
2. Increase concurrency gradually while monitoring AIQ and the target system.
3. Review whether bottlenecks come from SoapUI, the load generator, or the application.
4. Document results and any migration plan away from legacy SoapUI assets.

Last updated: October 7, 2026

Run Data-Enabled SoapUI Tests in AIQ

Recommended guidance

Run data-enabled SoapUI tests in AIQ only when the SoapUI asset remains required. For new API automation, consider whether Service Suite and DPLs can provide the same coverage with fewer legacy dependencies.

Before running

- Confirm SoapUI version, project file, data source, environment, endpoint, and credentials.
- Validate the data file or DPL values before execution.
- Check network, proxy, certificate, and firewall access from the AIQ host.
- Define expected pass/fail behavior for each data row.

Validation workflow

1. Run the SoapUI project with a small sample outside AIQ.
2. Run the AIQ wrapper with the same sample.
3. Review failures for malformed data, exhausted rows, endpoint issues, or assertion defects.
4. Document migration options if Service Suite can replace the SoapUI dependency.

Last updated: October 7, 2026