

Testing checklist: SDK or client library

Source: <https://howtotestaproduct.com/how-to-test/sdk/>

Generated: 2026-10-07. Licence: CC BY 4.0.

☐ 1. release smoke tests

Done when: Each release candidate carries its privacy manifest and a signature that Xcode validates, installs into sample host apps for every supported platform, builds from a clean cache, launches and completes the quick-start call.

Risks: dependency-failure, store-rejection. Tool class: mobile automation. Usually run by: QA engineer.

☐ 2. integration tests

Done when: A proxy capture of a sample host app shows the SDK sends only the data types in its privacy manifest, and nothing before the host app's consent call.

Risks: regulatory-penalty, data-breach. Tool class: proxy. Usually run by: developer.

☐ 3. compliance checks

Done when: The privacy manifest and the sample disclosure text list every data type the SDK collects and every recipient, signed off by legal, and the binary is signed.

Risks: regulatory-penalty, store-rejection. Tool class: test management. Usually run by: product manager.

☐ 4. security scanning

Done when: The dependency report for the release lists no known-malicious or high-severity package, every transitive dependency is pinned in the lockfile, and maintainer changes are reviewed.

Risks: data-breach. Tool class: dependency scanner. Usually run by: security specialist.

☐ 5. device and browser compatibility

Done when: The SDK builds and runs in sample host apps on the oldest and newest supported OS, language and package manager versions, and each iOS sample host app build passes App Store Connect validation with the SDK's privacy manifest.

Risks: store-rejection, dependency-failure. Tool class: device cloud. Usually run by: QA engineer.

☐ 6. unit tests

Done when: In the test report, every server-sent config and response field in the fixed sample set passes with a null, wrong-type and missing value and no crash.

Risks: dependency-failure. Tool class: unit framework. Usually run by: developer.

☐ 7. contract tests

Done when: Before any server config change goes live, staging responses match the schema that every supported SDK version expects.

Risks: dependency-failure. Tool class: contract tool. Usually run by: developer.

☐ 8. regression suite

Done when: The public API surface diff and the crash cases rerun on every release, and a removed or changed public method without a major version bump blocks the release.

Risks: dependency-failure. Tool class: test management. Usually run by: QA engineer.

☐ 9. resilience and failure testing

Done when: With the SDK backend returning errors, timeouts or malformed config, the sample host app keeps running and its own features work.

Risks: dependency-failure. Tool class: chaos tool. Usually run by: developer.