Associated Provider Notifications API Conformance Specification

The Aged Care Quality and Safety Commission's conformance requirements for providers and software operators

Version 1.0 | September 2026

Document ID: RBAC-1943816677-3669

Australian Government
Aged Care Quality and Safety Commission

Choice
Dignity
Respect

Disclaimer

The Aged Care Quality and Safety Commission (the Commission) makes the information and other material (Information) in this document available in good faith but without any representation or warranty as to its accuracy or completeness. The Commission cannot accept any responsibility for the consequences of any use of the Information. As the Information is of a general nature only, it is up to any person using or relying on the Information to make sure it is accurate, complete and suitable for them to use.

Document control

This document is maintained in electronic form and is uncontrolled in printed form. It is the responsibility of the user to verify that this copy is the latest revision.

Document information

Owner: Digital Branch, Corporate Division, Aged Care Quality and Safety Commission

Contact for enquiries

Aged Care Quality and Safety Commission

Phone: 1800 951 822

Email: apisupport@agedcarequality.gov.au

1. Purpose and scope

This specification sets out the requirements that providers and third-party software opperators must meet to achieve and maintain conformance with the Commission's Associated Provider Notifications Application Program Interface (API). Its purpose is to ensure that systems using the API can interact with it correctly, securely and reliably, and meet the Commission's requirements for production access.

It applies to every provider connecting to the API, whether they implement their own integration or use a third-party software operator. Providers and third-party software operators must demonstrate conformance to the specifications in this document to get production access.

The requirements have been developed with reference to Australian Government digital, API, cyber security and privacy standards and guidance, including the Digital Service Standard, the Australian Government API Standard, the Australian Government Information Security Manual, and the Privacy Act 1988 and Australian Privacy Principles, and are organised around the 2 components of Conformance described in Section 2.

2. How to read this specification

Each attribute has a unique ID, a short description, expected result, evidence required and a priority.

Conformance attributes are classified as:

- "Required" means a provider must meet the requirement to be considered conformant
- "Recommended" means the requirement is not mandatory now, but reflects good practice and may become required in a future version of this specification
- "Required where applicable" means the requirement only applies in specific circumstances, which are set out in the requirement text. This is distinct from "Required": where those circumstances do not apply to a given provider, the requirement does not apply at all.

Requirements are grouped under 2 components:

- technical and API Conformance, verified through a test submission (Section 3),
- security and Operational Conformance, self-attested by the Provider (Section 4).

Note: Conformance applies to the version of this specification and the API version against which the integration was assessed. Reassessment may be necessary if there are changes to the API, integration or operating environment .

3. Technical and API Conformance

Technical and API Conformance confirms that a provider's system can correctly call the API. It is verified through a test submission report before giving production access . The provider must give evidence that confirms the outcome of the test. The Commission will keep the evidence in the API Register.

ID	Attribute	Verification	Expected Result	Evidence	Priority
T-01	**Authentication** Authenticate with the API using the authentication method and client credentials specified by the Commission.	Execute authentication test scenarios using valid and invalid client credentials.	Valid credentials return an access token. Invalid credentials are rejected.	Test report containing request and response evidence.	Required
T-02	**Token management** Track and refresh access in accordance with the Commission's technical API requirements	Execute a test scenario that allows or simulates an access token approaching expiry while the system remains active, and confirm the system renews the access token automatically, without manual intervention.	The access token is renewed automatically before expiry, and subsequent requests continue to succeed without interruption or manual intervention.	Test report containing system or application logs showing the token renewal event and the timestamp of the subsequent successful request.	Required
T-03	**Credential management** Certificates used for authentication and secure communications are exchanged, validated and	Establish a connection to the Sandbox environment using an exchanged, valid public certificate, then attempt a connection using an invalid, expired or unrecognised certificate.	The connection succeeds when a valid, exchanged certificate is used, and is rejected when an invalid certificate is used.	Test report containing connection or handshake records for both the successful and rejected connection attempts.	Required

ID	Attribute	Verification	Expected Result	Evidence	Priority
	managed in accordance with Commission requirements.				
T-04	**Identifiers** Include the software and organisation identifiers the API expects in every submission.	Execute a submission and confirm the payload includes providerId and providerName, and, where applicable, authorisedGroupId.	The submission includes all required identifiers and is accepted with a 202 Accepted response.	Test report containing the submitted request payload.	Required
T-05	**Request structure** Structure submissions exactly as the API's published specification requires including mandatory field validation, schema compliance and payload size limits.	Execute and demonstrate test case scenarios covering valid payloads and known error scenarios including invalid field values and missing mandatory values in the payload	A valid submission is accepted with a 202 Accepted response; an invalid submission is rejected with a 400 Bad Request response, identifying the specific field that caused the error.	Test report containing the submitted request payloads and the corresponding 202 Accepted and 400 Bad Request responses received.	Required
T-06	**Response handling** Correctly process the API's response, including both success responses and error responses.	Submit a valid request and confirm the system correctly extracts the transactionId, correlationId and status from the response, then confirm the outcome via a subsequent GET request.	The system correctly parses all fields from the response, and the subsequent GET request confirms the change has been applied.	Test report containing the 202 Accepted response and the GET response confirming the applied change.	Required

ID	Attribute	Verification	Expected Result	Evidence	Priority
T-07	**Error messaging** Where an API response indicates a recoverable error, sufficient information must be presented to allow the user to understand the failure and take corrective action.	Using a rejected submission, confirm the system presents the error to the user, identifies the affected field using the **errorSource.parameter** value, and allows the error to be corrected and resubmitted.	The user is shown a clear, human-readable explanation of the error and the affected field, and can correct and resubmit the request successfully.	Test report containing a screenshot or export of the error message as presented to the user, and the successful response following resubmission.	Required
T-08	**Test evidence** Complete a successful test submission in the sandbox environment.	Execute the required conformance test scenarios set out in the provided Test Case Evidence Form, including multiple successful submissions, and complete the pass or fail outcome for each.	All required test cases are completed with a Pass outcome, and a completed evidence package is ready for submission to the Commission.	Completed Test Case Evidence Form with supporting evidence for each test case, submitted to the Commission.	Required
T-09	**Duplicate and retry handling** Manage retries and repeated requests in accordance with the API specification to minimise unintended duplicate submissions.	Confirmed through evidence gathered during executed test cases together with self-attestation of the provider's retry and idempotency handling. No dedicated Sandbox test case exists for this attribute.	Retries and repeated requests do not result in unintended duplicate submissions.	Self-attestation, supported by evidence from related test cases where available.	Required

Note: This component applies separately to every provider and third-party software operator integration, including where an operator is already conformant for other Business to Government (B2G) APIs. Section 6 sets out how this differs from Security and Operational Conformance by onboarding path.

4. Security and Operational Conformance

Security and Operational Conformance confirms that a provider's system meets the operational and security expectations set out below. The Commission cannot observe these controls from outside a provider's system, so this component is self-attested, not tested.

4.1 Identity and access management

ID	Requirement	Priority	Evidence
S-01	Role-based access. Restrict API functionality to authorised users, using role-based permissions.	Required	Self-attestation
S-02	Administrator separation. Privileged administrative access, such as user management and system configuration, must be separated from normal operational or data-submission activity.	Required	Self-attestation
S-03	User authentication. Require unique credentials and a minimum of password-based authentication meeting current ASD ISM complexity guidance, or an equivalent standards-based method (e.g. SAML, OIDC), for every user account.	Required	Self-attestation
S-04	Multi-factor authentication. Require multi-factor authentication for all administrative and privileged accounts associated with the API environment	Required	Self-attestation
S-05	Session timeout. End inactive sessions automatically, with a default of no more than 15 minutes and a configurable maximum of 2 hours.	Required	Self-attestation
S-06	Inactive account handling. Disable user accounts after an extended period of inactivity, no longer than 45 days.	Recommended	Self-attestation

ID	Requirement	Priority	Evidence
S-07	User record detail. Hold enough detail on each user account, such as name, identifier and role, to identify who performed an action.	Required	Self-attestation
S-08	Credential compromise management. Monitor for indicators of compromised user or API credentials, detect and investigate suspected compromise, and notify the Commission promptly if a compromise affecting the API integration is confirmed.	Required	Self-attestation

4.2 Audit and accountability

ID	Requirement	Priority
S-09	Activity logging. Record logon, logoff, credential change, failed authentication, access changes, privilege changes and other relevant administrative or security events for every user account.	Required
S-10	Submission logging. Record every submission made to the API, including timestamp, operation type, outcome, transaction and correlation ID and the submitting user.	Required
S-11	Log integrity. Protect these records against unauthorised modification.	Required
S-12	Log export. Retain these records for a minimum of 12 months and provide them in human-readable form if requested.	Required

4.3 Data protection and cryptography

ID	Requirement	Priority
S-13	Credential protection. Store credentials, private keys and access tokens securely and protect them from unauthorised access or disclosure. Credentials and private keys must not be stored in source code, configuration files, plain text, logs or other unsecured locations.	Required
S-14	Secure transport. Serve any web-facing part of the system over HTTPS with TLS 1.2 or higher	Required

ID	Requirement	Priority
S-15	Certificate validation. Validate certificates presented by external parties, including checking expiry, before trusting them.	Required

4.4 Secure development and operational resilience

ID	Requirement	Priority
S-16	Secure software development. Develop and maintain API integrations using secure practices consistent with the Australian Government ISM, including secure-by-design principles, vulnerability protection and appropriate security testing	Required
S-17	Patch management. Apply security patches and updates promptly and in a controlled way. Critical security patches must be assessed and applied within organisationally defined timeframes.	Required
S-18	Backup completeness. Where backup and restore functionality exists, ensure backups include all data, software and configuration needed to restore the system.	Recommended
S-19	Backup testing. Test backups through a full restoration at least once, then on a regular basis.	Recommended
S-20	Backup access control. Restrict access to backup files to authorised administrators.	Recommended
S-21	Unauthorised code execution. Do not allow unauthorised execution of scripts, installers or executables from within the system.	Required
S-22	Vulnerability Management. Known vulnerabilities are identified, assessed and remediated in a risk-based manner.	Required

4.5 Privacy and information handling

- Purpose limitation: Use information obtained or submitted through the API only for authorised purposes
- Data minimisation: Collect and transmit only the information necessary for the relevant API operation

- Privacy obligations: Collect, use, store and disclose personal information in accordance with applicable privacy obligations and Commission requirements
- Retention and disposal: Retain and securely dispose of API information in accordance with applicable legal, contractual and Commission requirements
- Privacy incident management: Maintain arrangements for identifying, managing and reporting suspected unauthorised access, disclosure or loss of information.

5. How conformance is confirmed

Technical and API Conformance

Conformance is confirmed through a successful test submission, with evidence recorded in the API Register. During the registration process, providers will need to execute test scenarios in the Sandbox environment before getting access to production. They will need to share test evidence reports with the Commission to verify that technical conformance has been achieved. During onboarding, providers will be given test evidence templates to complete and return to us to confirm their technical conformance.

Security and operational conformance

A provider confirms they align with Section 4 (Security and Operational Conformance) through a single declaration on the registration form.

The Commission does not collect supporting evidence for security and operational conformance upfront. The Commission may request it at any time, for example as a spot-check, following an incident, or where something identified during technical testing suggests an inconsistency with what has been declared.

6. By onboarding path

This specification applies to every provider, whether they implement their own integration or use a third-party software operator. Both paths must meet Technical and API Conformance in full. Security and Operational Conformance differs, as set out below.

Component	Self-implementing provider	Provider using a third-party software operator
Technical and API Conformance	Full test required.	Can rely on the Conformance evidence provided and established by the third-party at the time of their registration
Security and Operational Conformance	Full declaration required.	Can rely on the declaration and any provided evidence gathered as part of the third-party software agreement

7. When conformance needs to be reassessed

Provider must notify the Commission, and conformance is reassessed, when a material change can potentially impact the connectivity or use of the API. Examples of a material change are:

- a change of hosting environment for the provider's system
- a change to the authentication method used to connect to the API,
- a change of software vendor
- a significant change to identity and access management architecture
- a change to multi-factor authentication implementation,
- a security control failure
- a major vulnerability affecting the API integration.

Note: These triggers apply to both Technical and API Conformance and Security and Operational Conformance.

Event	Treatment
Major API version	Technical re-conformance
Minor backwards-compatible version	Attestation or targeted testing
Authentication mechanism changes	Targeted technical re-conformance

Event	Treatment
Hosting or security architecture materially changes	Security reassessment
Provider changes software operator	New integration or onboarding conformance
Material software architecture change	Risk-based reassessment
Security incident affecting integration	Security reassessment
Credential compromise	Credential replacement and reassessment where necessary
Material vulnerability affecting integration	Risk-based reassessment
Evidence indicates non-conformance	Targeted or full reassessment
Conformance specification materially changes	Defined transition or reassessment process
Periodic expiry	**24 months**, subject to Commission risk decision

8. What happens if conformance is not met

Technical and API Conformance

The provider remains in 'Testing in Progress' stage and can remediate and retry. They don't need to restart the registration process.

Security and Operational Conformance

Registration is blocked, and the outcome is recorded in the API Register. This applies where a declaration is false, or where evidence is not enough.

9. Definitions

Provider: An organisation registered to provide government subsidised aged care, registered or registering to submit data through the API, whether directly or through a third party software operator.

Self-implementing provider: A provider that builds or operates its own integration with the API, rather than using a third-party software operator.

Third-party software pperator: An organisation that submits API data on behalf of one or more providers, using its own software instance.

Client ID: Part of the Client Credentials (Client ID and Client Secret) issued to a provider once eligibility and conformance have both been confirmed, enabling production access.

API Register: The Commission's record of provider and third-party software operator registration, eligibility and conformance status.

Integration Team: The Commission team that verifies Technical and API Conformance and requests evidence of Security and Operational Conformance.

Australian Government digital service standards: https://www.digital.gov.au/policy/digital-experience/digital-service-standard

Australian Government API Standard: https://api.gov.au/

Australian Government Information Security Manual (ISM): https://www.cyber.gov.au/ism

OAIC Australian Privacy Principles guidelines: https://www.oaic.gov.au/privacy/australian-privacy-principles/australian-privacy-principles-guidelines

10. Document control

Owner: Digital Branch, Corporate Division, Aged Care Quality and Safety Commission

Review date: this specification will be reviewed on a regular basis, and whenever the Conformance model changes.

Version	Date	Summary of change
1.0	Sep 2026	Initial release of the API

Australian Government
Aged Care Quality and Safety Commission

Choice
Dignity
Respect

Phone
1800 951 822

Web
agedcarequality.gov.au

Write
Aged Care Quality and Safety Commission
GPO Box 9819, in your capital city