

Developer Guide and API Reference Template

Use this guide to help developers reach a first successful API call, complete integration tasks, and look up exact interface behavior. Keep learning instructions separate from the endpoint and schema references they use.

Complete the bracketed fields with approved information for the product, organization, or program being documented. Repeat the named forms where several tasks or subjects need separate records. Remove sections that do not apply and retain links to the material used to complete the document.

Document control	Entry
Document identifier	[Identifier used by your team]
Scope and applicable version	[Product, program, locations, and version covered]
Prepared and maintained by	[Responsible role or team]
Revision and review	[Revision, review date, and approval reference]

Overview and Getting Started

Explain what the API enables and provide a short path to a verifiable first result. Keep the environment and API version explicit throughout the quickstart.

Field	Complete with
API purpose	[Supported use cases, audience, product version, and scope]
Environments	[Base URL, environment purpose, version, and access prerequisites]
First result	[Quickstart goal, required account or tools, and expected response]
Learning path	[Recommended tutorials, task guides, concepts, and reference entry points]

Repeat this procedure for each supported task or variant. Place the applicable warning before the affected action and record a result that can be checked.

Step	Action	Expected result	Evidence
1	[One action using the exact interface label or approved command]	[Observable state]	[Check or record]
2	[Next action in dependency order]	[Observable state]	[Check or record]
3	[Verification or handover action]	[Completion condition]	[Retained evidence]

Authentication and Authorization

Describe each supported authentication method separately. Reference credentials by purpose and approved storage location rather than placing live secrets in the guide.

Field	Complete with
Authentication scheme	[Scheme name, applicable clients, and credential acquisition reference]
Permissions	[Scope or role, permitted operations, and access approval path]
Request authentication	[Header or token placement and a redacted request example reference]
Credential lifecycle	[Refresh, rotation, expiry, revocation, and failure behavior]

Core Concepts

Repeat the concept record for conventions that apply across endpoints. Give examples only after the convention and its applicability are clear.

Field	Complete with
Concept	[Resource model, pagination, rate limits, idempotency, or another convention]
Rule and scope	[Expected behavior and the operations to which it applies]
Client handling	[What a client must send, store, or do in response]
Examples and limits	[Supported example reference, edge cases, and documented limits]

How To Guides

Create one guide per developer goal and link to the endpoint reference instead of duplicating its full parameter tables. Keep prerequisite setup separate from the task itself.

Field	Complete with
Goal	[One integration outcome and the audience or client type]
Prerequisites	[Access, data, configuration, dependencies, and relevant concepts]
Operations used	[Endpoint or SDK method references in execution order]
Verification	[Expected response or state, error recovery, and evidence of completion]

Repeat this procedure for each supported task or variant. Place the applicable warning before the affected action and record a result that can be checked.

Step	Action	Expected result	Evidence
1	[One action using the exact interface label or approved command]	[Observable state]	[Check or record]
2	[Next action in dependency order]	[Observable state]	[Check or record]
3	[Verification or handover action]	[Completion condition]	[Retained evidence]

API Reference

Repeat this operation record for every supported endpoint. Preserve exact field names, types, formats, and response behavior from the authoritative specification.

Field	Complete with
Operation	[HTTP method, path, operation identifier, summary, and resource group]
Request	[Path, query, header, and body fields with type, requirement, constraints, and example]
Responses	[Status code, headers, schema, example, and success meaning]
Errors and permissions	[Error response, relevant scope, retry behavior, and source reference]

Data Models

Keep reusable schemas in one place and refer to them from endpoint records. Distinguish an omitted field from a field whose value may be null.

Field	Complete with
Schema	[Name, purpose, version, and operations that use it]
Field dictionary	[Exact field name, type, format, required or nullable status, and meaning]
Validation	[Allowed values, length or range, units, and relationships between fields]
Compatibility	[Change rules, deprecated fields, and migration reference]

Webhooks and Events

Document subscription setup separately from the event payload reference. Describe verification and delivery handling using the behavior the provider supports.

Field	Complete with
Subscription	[Supported event family, subscription task, callback requirements, and permissions]
Event payload	[Event identifier, schema, fields, redacted example, and version]
Verification	[Signature or authenticity check and the relevant source reference]
Delivery handling	[Acknowledgment, retries, duplicates, ordering, and failure investigation]

SDKs and Tools

Repeat the setup record for each supported language or tool. Keep client method names aligned with the documented package version.

Field	Complete with
SDK or tool	[Package name, supported runtime, version, and installation source]
Setup	[Installation, configuration, credential reference, and environment selection]
First use	[Minimal task, expected result, and code example reference]
Reference	[Client method or tool command documentation, limits, and support channel]

Repeat this procedure for each supported task or variant. Place the applicable warning before the affected action and record a result that can be checked.

Step	Action	Expected result	Evidence
1	[One action using the exact interface label or approved command]	[Observable state]	[Check or record]
2	[Next action in dependency order]	[Observable state]	[Check or record]
3	[Verification or handover action]	[Completion condition]	[Retained evidence]

Errors and Troubleshooting

Group errors by their observable response and document how a developer distinguishes the underlying causes. Keep diagnostic requests and retry guidance traceable to the specification.

Field	Complete with
Error contract	[Status or error code, message format, and diagnostic fields]
Likely conditions	[Documented causes and the checks that distinguish them]
Client response	[Correct input, access change, retry, or escalation action]
Recovery evidence	[Successful follow-up result, request identifier, and support information]

Versioning Changelog and Glossary

Record changes by API version and make breaking changes easy to find. Define specialized terms consistently with the operation and model references.

Field	Complete with
Version entry	[Version or release identifier, date if known, and supported scope]
Change	[Added, changed, deprecated, or removed behavior and affected operations]
Migration	[Required client action, compatibility period if documented, and migration guide]
Terminology	[Term, definition, and references to its use in the guide]