

Common audit log format for Google Cloud Platform API operations.

JSON representation

Fields

Fields	
serviceName	string The name of the API service performing the operation. For example, "datastore.googleapis.com".
methodName	string The name of the service method or operation. For API calls, this should be the name of the API method. For example,
resourceName	string The resource or collection that is the target of the operation. The name is a scheme-less URI, not including the API service name. For example:
numResponseItems	string (<u>int64</u> (https://developers.google.com/discovery/v1/type-format) format) The number of items returned from a List or Query API method, if applicable.
status	object (<u>Status</u> (/logging/docs/reference/audit/auditlog/rest/Shared.Types/AuditLog#Status)) The status of the overall operation.

Fields	
authenticationInfo	<p>object (<u>AuthenticationInfo</u> (/logging/docs/reference/audit/auditlog/rest/Shared.Types/AuditLog#AuthenticationInfo))</p> <p>Authentication information.</p>
authorizationInfo[]	<p>object (<u>AuthorizationInfo</u> (/logging/docs/reference/audit/auditlog/rest/Shared.Types/AuditLog#AuthorizationInfo))</p> <p>Authorization information. If there are multiple resources or permissions involved, then there is one AuthorizationInfo element for each {resource, permission} tuple.</p>
requestMetadata	<p>object (<u>RequestMetadata</u> (/logging/docs/reference/audit/auditlog/rest/Shared.Types/AuditLog#RequestMetadata))</p> <p>Metadata about the operation.</p>
request	<p>object (<u>Struct</u> (https://developers.google.com/protocol-buffers/docs/reference/google.protobuf#google.protobuf.Struct) format)</p> <p>The operation request. This may not include all request parameters, such as those that are too large, privacy-sensitive, or duplicated elsewhere in the log record. It should never include user-generated data, such as file contents. When the JSON object represented here has a proto equivalent, the proto name will be indicated in the @type property.</p>
response	<p>object (<u>Struct</u> (https://developers.google.com/protocol-buffers/docs/reference/google.protobuf#google.protobuf.Struct) format)</p> <p>The operation response. This may not include all response elements, such as those that are too large, privacy-sensitive, or duplicated elsewhere in the log record. It should never include user-generated data, such as file contents. When the JSON object represented here has a proto equivalent, the proto name will be indicated in the @type property.</p>

Fields

serviceData

object

Other service-specific data about the request, response, and other activities.

An object containing fields of an arbitrary type. An additional field `"@type"` contains a URI identifying the type. Example: `{ "id": 1234, "@type": "types.example.com/standard/id" }`.

The `Status` type defines a logical error model that is suitable for different programming environments, including REST APIs and RPC APIs. It is used by `gRPC` (<https://github.com/grpc>). The error model is designed to be:

- Simple to use and understand for most users
- Flexible enough to meet unexpected needs

The `Status` message contains three pieces of data: error code, error message, and error details. The error code should be an enum value of `google.rpc.Code`, but it may accept additional error codes if needed. The error message should be a developer-facing English message that helps developers *understand* and *resolve* the error. If a localized user-facing error message is needed, put the localized message in the error details or localize it in the client. The optional error details may contain arbitrary information about the error. There is a predefined set of error detail types in the package `google.rpc` which can be used for common error conditions.

The `Status` message is the logical representation of the error model, but it is not necessarily the actual wire format. When the `Status` message is exposed in different client libraries and different wire protocols, it can be mapped differently. For example, it will likely be mapped to some exceptions in Java, but more likely mapped to some error codes in C.

The error model and the `Status` message can be used in a variety of environments, either with or without APIs, to provide a consistent developer experience across different environments.

Example uses of this error model include:

- **Partial errors.** If a service needs to return partial errors to the client, it may embed the `Status` in the normal response to indicate the partial errors.
- **Workflow errors.** A typical workflow has multiple steps. Each step may have a `Status` message for error reporting purpose.
- **Batch operations.** If a client uses batch request and batch response, the `Status` message should be used directly inside batch response, one for each error sub-response.
- **Asynchronous operations.** If an API call embeds asynchronous operation results in its response, the status of those operations should be represented directly using the `Status` message.
- **Logging.** If some API errors are stored in logs, the message `Status` could be used directly after any stripping needed for security/privacy reasons.

JSON representation

Fields

code

number

The status code, which should be an enum value of `google.rpc.Code`.

Fields

message

string

A developer-facing error message, which should be in English. Any user-facing error message should be localized and sent in the [google.rpc.Status.details](#) (/logging/docs/reference/audit/auditlog/rest/Shared.Types/AuditLog#Status.FIELDS.details) field, or localized by the client.

details[]

object

A list of messages that carry the error details. There will be a common set of message types for APIs to use.

An object containing fields of an arbitrary type. An additional field "@type" contains a URI identifying the type. Example: { "id": 1234, "@type": "types.example.com/standard/id" }.

Authentication information for the operation.

JSON representation

Fields

principalEmail

string

The email address of the authenticated user making the request.

authoritySelector

string

The authority selector specified by the requestor, if any. It is not guaranteed that the principal was allowed to use this authority.

Authorization information for the operation.

JSON representation

Fields

resource	string The resource being accessed, as a REST-style string. For example:
permission	string The required IAM permission.
granted	boolean Whether or not authorization for resource and permission was granted

Metadata about the request.

JSON representation

Fields	
callerIp	string The IP address of the caller.
callerSuppliedUserAgent	string The user agent of the caller. This information is not authenticated and should be treated accordingly. For example: <ul style="list-style-type: none">• google-api-python-client/1.4.0: The request was made by the Google API client for Python.• Cloud SDK Command Line Tool apitools-client/1.0 gcloud/0.9.62: The request was made by the Google Cloud SDK CLI (gcloud).• AppEngine-Google; (+http://code.google.com/appengine; appid: s~my-project: The request was made from the my-project App Engine app.