

# Get specific versions of resources

Every resource has a version field that changes every time the resource changes — the `etag` field. Etags are a standard part of HTTP and are supported in the calendar API for two cases:

- on resource modifications to ensure that there has been no other write to this resource in the meantime (conditional modification)
- on resource retrieval to only retrieve resource data if the resource has changed (conditional retrieval)

## Conditional modification

If you want to update or delete a resource only if it has not changed since you last retrieved it, you can specify an `If-Match` header that contains the value of the etag from the previous retrieval. This is very useful to prevent lost modifications on resources. The clients have the option of re-retrieving the resource and re-applying the changes.

If the entry (and its etag) has not changed since the last retrieval, the modification succeeds and the new version of the resource with the new etag is returned. Otherwise, you will get a 412 (Precondition failed) response code.

**Important:** There is no support for conditional modifications for insert operations. Instead, it is guaranteed that if you are allowed to provide a resource ID, then the operation will only succeed if no existing entry has that ID.

The snippet of sample code below demonstrates how to perform conditional modifications with the [Java client library](#) (/api-client-library/java/apis/calendar/v3).

```
private static void run() throws IOException {
    // Create a test event.
    Event event = Utils.createTestEvent(client, "Test Event");
    System.out.println(String.format("Event created: %s", event.getHtmlLink()));

    // Pause while the user modifies the event in the Calendar UI.
    System.out.println("Modify the event's description and hit enter to continue
```

```

System.in.read();

// Modify the local copy of the event.
event.setSummary("Updated Test Event");

// Update the event, making sure that we don't overwrite other changes.
int numAttempts = 0;
boolean isUpdated = false;
do {
    Calendar.Events.Update request = client.events().update("primary", event.g
    request.setRequestHeaders(new HttpHeaders().setIfMatch(event.getEtag()));
    try {
        event = request.execute();
        isUpdated = true;
    } catch (GoogleJsonResponseException e) {
        if (e.getStatusCode() == 412) {
            // A 412 status code, "Precondition failed", indicates that the etag v
            // match, and the event was updated on the server since we last retriee
            // {@link Calendar.Events.Get} to retrieve the latest version.
            Event latestEvent = client.events().get("primary", event.getId()).exec

            // You may want to have more complex logic here to resolve conflicts.
            // simply overwriting the summary.
            latestEvent.setSummary(event.getSummary());
            event = latestEvent;
        } else {
            throw e;
        }
    }
    numAttempts++;
} while (!isUpdated && numAttempts <= MAX_UPDATE_ATTEMPTS);

if (isUpdated) {
    System.out.println("Event updated.");
} else {
    System.out.println(String.format("Failed to update event after %d attempts
}
}
}

```

/nc/src/main/java/com/google/api/services/samples/calendar/sync/ConditionalModificationSample.java)

## Conditional retrieval

If you want to retrieve a resource only if it has changed since you last retrieved it, you can specify an **If-None-Match** header which contains the value of the etag from the previous retrieval. If the entry (and thus its etag) has changed since the last retrieval, the new version of the resource with the new etag will be returned. Otherwise you will get a 304 (Not Modified) response code.

**Note:** There are special cases in which etags won't change, such as when one of the read-only fields of a `calendarList` entry changes (e.g., calendar properties or ACLs).

The snippet of sample code below demonstrates how to perform conditional retrieval with the Java client library ([/api-client-library/java/apis/calendar/v3](#)).

```
private static void run() throws IOException {
    // Create a test event.
    Event event = Utils.createTestEvent(client, "Test Event");
    System.out.println(String.format("Event created: %s", event.getHtmlLink()));

    // Pause while the user modifies the event in the Calendar UI.
    System.out.println("Modify the event's description and hit enter to continue");
    System.in.read();

    // Fetch the event again if it's been modified.
    Calendar.Events.Get getRequest = client.events().get("primary", event.getId());
    getRequest.setRequestHeaders(new HttpHeaders().setIfNoneMatch(event.getEtag()));
    try {
        event = getRequest.execute();
        System.out.println("The event was modified, retrieved latest version.");
    } catch (GoogleJsonResponseException e) {
        if (e.getStatusCode() == 304) {
            // A 304 status code, "Not modified", indicates that the etags match, and
            // not been modified since we last retrieved it.
            System.out.println("The event was not modified, using local version.");
        } else {
            throw e;
        }
    }
}
```

`src/main/java/com/google/api/services/samples/calendar/sync/ConditionalRetrievalSample.java`

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2025-07-30 UTC.