

Google API Documentation

classroom API Documentation

Version: v1

Manages classes, rosters, and invitations in Google Classroom.

Google API Documentation

Endpoint: invitations.list

HTTP Method: GET

Path: v1/invitations

Description: Returns a list of invitations that the requesting user is permitted to view, restricted to those that match the list request. **Note:* At least one of `user_id` or `course_id` must be supplied. Both fields can be supplied. This method returns the following error codes: ** `PERMISSION_DENIED` for access errors.*

Google API Documentation

Endpoint: invitations.delete

HTTP Method: DELETE

Path: v1/invitations/{id}

Description: Deletes an invitation. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting user is not permitted to delete the requested invitation or for access errors. * `NOT_FOUND` if no invitation exists with the requested ID.

Google API Documentation

Endpoint: invitations.get

HTTP Method: GET

Path: v1/invitations/{id}

Description: Returns an invitation. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting user is not permitted to view the requested invitation or for access errors. * `NOT_FOUND` if no invitation exists with the requested ID.

Google API Documentation

Endpoint: invitations.accept

HTTP Method: POST

Path: v1/invitations/{id}:accept

Description: Accepts an invitation, removing it and adding the invited user to the teachers or students (as appropriate) of the specified course. Only the invited user may accept an invitation. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting user is not permitted to accept the requested invitation or for access errors. * `FAILED_PRECONDITION` for the following request errors: * CourseMemberLimitReached * CourseNotModifiable * CourseTeacherLimitReached * UserGroupsMembershipLimitReached * `NOT_FOUND` if no invitation exists with the requested ID.

Google API Documentation

Endpoint: invitations.create

HTTP Method: POST

Path: v1/invitations

Description: Creates an invitation. Only one invitation for a user and course may exist at a time. Delete and re-create an invitation to make changes. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting user is not permitted to create invitations for this course or for access errors. * `NOT_FOUND` if the course or the user does not exist. * `FAILED_PRECONDITION`: * if the requested user's account is disabled. * if the user already has this role or a role with greater permissions. * for the following request errors: * IneligibleOwner * `ALREADY_EXISTS` if an invitation for the specified user and course already exists.

Google API Documentation

Endpoint: `courses.list`

HTTP Method: GET

Path: `v1/courses`

Description: Returns a list of courses that the requesting user is permitted to view, restricted to those that match the request. Returned courses are ordered by creation time, with the most recently created coming first. This method returns the following error codes: * `PERMISSION_DENIED` for access errors. * `INVALID_ARGUMENT` if the query argument is malformed. * `NOT_FOUND` if any users specified in the query arguments do not exist.

Google API Documentation

Endpoint: `courses.get`

HTTP Method: GET

Path: `v1/courses/{id}`

Description: Returns a course. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course or for access errors. * ``NOT_FOUND`` if no course exists with the requested ID.

Google API Documentation

Endpoint: `courses.delete`

HTTP Method: DELETE

Path: `v1/courses/{id}`

Description: Deletes a course. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to delete the requested course or for access errors. * ``NOT_FOUND`` if no course exists with the requested ID.

Google API Documentation

Endpoint: `courses.patch`

HTTP Method: PATCH

Path: `v1/courses/{id}`

Description: Updates one or more fields in a course. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to modify the requested course or for access errors. * ``NOT_FOUND`` if no course exists with the requested ID. * ``INVALID_ARGUMENT`` if invalid fields are specified in the update mask or if no update mask is supplied. * ``FAILED_PRECONDITION`` for the following request errors: * `CourseNotModifiable` * `InactiveCourseOwner` * `IneligibleOwner`

Google API Documentation

Endpoint: `courses.create`

HTTP Method: POST

Path: `v1/courses`

Description: Creates a course. The user specified in `ownerId` is the owner of the created course and added as a teacher. A non-admin requesting user can only create a course with themselves as the owner. Domain admins can create courses owned by any user within their domain. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting user is not permitted to create courses or for access errors. * `NOT_FOUND` if the primary teacher is not a valid user. * `FAILED_PRECONDITION` if the course owner's account is disabled or for the following request errors: * `UserCannotOwnCourse` * `UserGroupsMembershipLimitReached` * `ALREADY_EXISTS` if an alias was specified in the `id` and already exists.

Google API Documentation

Endpoint: courses.update

HTTP Method: PUT

Path: v1/courses/{id}

Description: Updates a course. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting user is not permitted to modify the requested course or for access errors. * `NOT_FOUND` if no course exists with the requested ID. * `FAILED_PRECONDITION` for the following request errors: * CourseNotModifiable

Google API Documentation

Endpoint: `courses.courseWork.get`

HTTP Method: GET

Path: `v1/courses/{courseId}/courseWork/{id}`

Description: Returns course work. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course or course work, or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course or course work does not exist.

Google API Documentation

Endpoint: `courses.courseWork.getAddOnContext`

HTTP Method: GET

Path: `v1/courses/{courseId}/courseWork/{itemId}/addOnContext`

Description: Gets metadata for Classroom add-ons in the context of a specific post. To maintain the integrity of its own data and permissions model, an add-on should call this to validate query parameters and the requesting user's role whenever the add-on is opened in an [iframe](<https://developers.google.com/classroom/add-ons/get-started/iframes/iframes-overview>). This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.courseWork.patch`

HTTP Method: PATCH

Path: `v1/courses/{courseId}/courseWork/{id}`

Description: Updates one or more fields of a course work. See `google.classroom.v1.CourseWork` for details of which fields may be updated and who may change them. This request must be made by the Developer Console project of the [OAuth client ID](<https://support.google.com/cloud/answer/6158849>) used to create the corresponding course work item. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting developer project did not create the corresponding course work, if the user is not permitted to make the requested modification to the student submission, or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``FAILED_PRECONDITION`` if the requested course work has already been deleted. * ``NOT_FOUND`` if the requested course or course work does not exist.

Google API Documentation

Endpoint: `courses.courseWork.updateRubric`

HTTP Method: PATCH

Path: `v1/courses/{courseId}/courseWork/{courseWorkId}/rubric`

Description: Updates a rubric. See `google.classroom.v1.Rubric` for details of which fields can be updated. Rubric update capabilities are [limited](/classroom/rubrics/limitations) once grading has started. The requesting user and course owner must have rubrics creation capabilities. For details, see [licensing requirements](https://developers.google.com/classroom/rubrics/limitations#license-requirements). This request must be made by the Google Cloud console of the [OAuth client ID](https://support.google.com/cloud/answer/6158849) used to create the parent course work item. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting developer project didn't create the corresponding course work, if the user isn't permitted to make the requested modification to the rubric, or for access errors. This error code is also returned if grading has already started on the rubric. * ``INVALID_ARGUMENT`` if the request is malformed and for the following request error: * ``RubricCriteriaInvalidFormat`` * ``NOT_FOUND`` if the requested course, course work, or rubric doesn't exist or if the user doesn't have access to the corresponding course work. * ``INTERNAL`` if grading has already started on the rubric.

Google API Documentation

Endpoint: `courses.courseWork.delete`

HTTP Method: DELETE

Path: `v1/courses/{courseId}/courseWork/{id}`

Description: Deletes a course work. This request must be made by the Developer Console project of the [OAuth client ID](<https://support.google.com/cloud/answer/6158849>) used to create the corresponding course work item. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting developer project did not create the corresponding course work, if the requesting user is not permitted to delete the requested course or for access errors. * ``FAILED_PRECONDITION`` if the requested course work has already been deleted. * ``NOT_FOUND`` if no course exists with the requested ID.

Google API Documentation

Endpoint: `courses.courseWork.list`

HTTP Method: GET

Path: `v1/courses/{courseId}/courseWork`

Description: Returns a list of course work that the requester is permitted to view. Course students may only view `PUBLISHED` course work. Course teachers and domain administrators may view all course work. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting user is not permitted to access the requested course or for access errors. * `INVALID_ARGUMENT` if the request is malformed. * `NOT_FOUND` if the requested course does not exist.

Google API Documentation

Endpoint: `courses.courseWork.modifyAssignees`

HTTP Method: POST

Path: `v1/courses/{courseId}/courseWork/{id}:modifyAssignees`

Description: Modifies assignee mode and options of a coursework. Only a teacher of the course that contains the coursework may call this method. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course or course work or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course or course work does not exist.

Google API Documentation

Endpoint: `courses.courseWork.create`

HTTP Method: POST

Path: `v1/courses/{courseId}/courseWork`

Description: Creates course work. The resulting course work (and corresponding student submissions) are associated with the Developer Console project of the [OAuth client ID](<https://support.google.com/cloud/answer/6158849>) used to make the request. Classroom API requests to modify course work and student submissions must be made with an OAuth client ID from the associated Developer Console project. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course, create course work in the requested course, share a Drive attachment, or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course does not exist. * ``FAILED_PRECONDITION`` for the following request error: * AttachmentNotVisible

Google API Documentation

Endpoint: `courses.courseWork.addOnAttachments.patch`

HTTP Method: PATCH

Path: `v1/courses/{courseId}/courseWork/{itemId}/addOnAttachments/{attachmentId}`

Description: Updates an add-on attachment. Requires the add-on to have been the original creator of the attachment. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.courseWork.addOnAttachments.get`

HTTP Method: GET

Path: `v1/courses/{courseId}/courseWork/{itemId}/addOnAttachments/{attachmentId}`

Description: Returns an add-on attachment. Requires the add-on requesting the attachment to be the original creator of the attachment. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.courseWork.addOnAttachments.list`

HTTP Method: GET

Path: `v1/courses/{courseId}/courseWork/{itemId}/addOnAttachments`

Description: Returns all attachments created by an add-on under the post. Requires the add-on to have active attachments on the post or have permission to create new attachments on the post. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.courseWork.addOnAttachments.create`

HTTP Method: POST

Path: `v1/courses/{courseId}/courseWork/{itemId}/addOnAttachments`

Description: Creates an add-on attachment under a post. Requires the add-on to have permission to create new attachments on the post. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.courseWork.addOnAttachments.delete`

HTTP Method: DELETE

Path: `v1/courses/{courseId}/courseWork/{itemId}/addOnAttachments/{attachmentId}`

Description: Deletes an add-on attachment. Requires the add-on to have been the original creator of the attachment. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.courseWork.addOnAttachments.studentSubmissions.get`

HTTP Method: GET

Path: `v1/courses/{courseId}/courseWork/{itemId}/addOnAttachments/{attachmentId}/studentSubmissions/{submissionId}`

Description: Returns a student submission for an add-on attachment. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.courseWork.addOnAttachments.studentSubmissions.patch`

HTTP Method: PATCH

Path: `v1/courses/{courseId}/courseWork/{itemId}/addOnAttachments/{attachmentId}/studentSubmissions/{submissionId}`

Description: Updates data associated with an add-on attachment submission. Requires the add-on to have been the original creator of the attachment and the attachment to have a positive `max_points` value set. This method returns the following error codes: * `PERMISSION_DENIED` for access errors. * `INVALID_ARGUMENT` if the request is malformed. * `NOT_FOUND` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.courseWork.rubrics.create`

HTTP Method: POST

Path: `v1/courses/{courseId}/courseWork/{courseWorkId}/rubrics`

Description: Creates a rubric. The requesting user and course owner must have rubrics creation capabilities. For details, see [licensing requirements](https://developers.google.com/classroom/rubrics/limitations#license-requirements). For further details, see [Rubrics structure and known limitations](/classroom/rubrics/limitations). This request must be made by the Google Cloud console of the [OAuth client ID](https://support.google.com/cloud/answer/6158849) used to create the parent course work item. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user isn't permitted to create rubrics for course work in the requested course. * ``INTERNAL`` if the request has insufficient OAuth scopes. * ``INVALID_ARGUMENT`` if the request is malformed and for the following request error: * ``RubricCriteriaInvalidFormat`` * ``NOT_FOUND`` if the requested course or course work don't exist or the user doesn't have access to the course or course work. * ``FAILED_PRECONDITION`` for the following request error: * ``AttachmentNotVisible``

Google API Documentation

Endpoint: `courses.courseWork.rubrics.get`

HTTP Method: GET

Path: `v1/courses/{courseId}/courseWork/{courseWorkId}/rubrics/{id}`

Description: Returns a rubric. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course, course work, or rubric doesn't exist or if the user doesn't have access to the corresponding course work.

Google API Documentation

Endpoint: `courses.courseWork.rubrics.delete`

HTTP Method: DELETE

Path: `v1/courses/{courseId}/courseWork/{courseWorkId}/rubrics/{id}`

Description: Deletes a rubric. The requesting user and course owner must have rubrics creation capabilities. For details, see [licensing requirements](https://developers.google.com/classroom/rubrics/limitations#license-requirements). This request must be made by the Google Cloud console of the [OAuth client ID](https://support.google.com/cloud/answer/6158849) used to create the corresponding rubric. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting developer project didn't create the corresponding rubric, or if the requesting user isn't permitted to delete the requested rubric. * `NOT_FOUND` if no rubric exists with the requested ID or the user does not have access to the course, course work, or rubric. * `INVALID_ARGUMENT` if grading has already started on the rubric.

Google API Documentation

Endpoint: `courses.courseWork.rubrics.list`

HTTP Method: GET

Path: `v1/courses/{courseId}/courseWork/{courseWorkId}/rubrics`

Description: Returns a list of rubrics that the requester is permitted to view. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course or course work doesn't exist or if the user doesn't have access to the corresponding course work.

Google API Documentation

Endpoint: `courses.courseWork.rubrics.patch`

HTTP Method: PATCH

Path: `v1/courses/{courseId}/courseWork/{courseWorkId}/rubrics/{id}`

Description: Updates a rubric. See `google.classroom.v1.Rubric` for details of which fields can be updated. Rubric update capabilities are [limited](/classroom/rubrics/limitations) once grading has started. The requesting user and course owner must have rubrics creation capabilities. For details, see [licensing requirements](https://developers.google.com/classroom/rubrics/limitations#license-requirements). This request must be made by the Google Cloud console of the [OAuth client ID](https://support.google.com/cloud/answer/6158849) used to create the parent course work item. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting developer project didn't create the corresponding course work, if the user isn't permitted to make the requested modification to the rubric, or for access errors. This error code is also returned if grading has already started on the rubric. * ``INVALID_ARGUMENT`` if the request is malformed and for the following request error: * ``RubricCriteriaInvalidFormat`` * ``NOT_FOUND`` if the requested course, course work, or rubric doesn't exist or if the user doesn't have access to the corresponding course work. * ``INTERNAL`` if grading has already started on the rubric.

Google API Documentation

Endpoint: `courses.courseWork.studentSubmissions.list`

HTTP Method: GET

Path: `v1/courses/{courseId}/courseWork/{courseWorkId}/studentSubmissions`

Description: Returns a list of student submissions that the requester is permitted to view, factoring in the OAuth scopes of the request. ``-`` may be specified as the ``course_work_id`` to include student submissions for multiple course work items. Course students may only view their own work. Course teachers and domain administrators may view all student submissions. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course or course work, or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course does not exist.

Google API Documentation

Endpoint: `courses.courseWork.studentSubmissions.reclaim`

HTTP Method: POST

Path: `v1/courses/{courseId}/courseWork/{courseWorkId}/studentSubmissions/{id}:reclaim`

Description: Reclaims a student submission on behalf of the student that owns it. Reclaiming a student submission transfers ownership of attached Drive files to the student and updates the submission state. Only the student that owns the requested student submission may call this method, and only for a student submission that has been turned in. This request must be made by the Developer Console project of the [OAuth client ID](<https://support.google.com/cloud/answer/6158849>) used to create the corresponding course work item. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course or course work, unsubmit the requested student submission, or for access errors. * ``FAILED_PRECONDITION`` if the student submission has not been turned in. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course, course work, or student submission does not exist.

Google API Documentation

Endpoint: `courses.courseWork.studentSubmissions.modifyAttachments`

HTTP Method: POST

Path: `v1/courses/{courseId}/courseWork/{courseWorkId}/studentSubmissions/{id}:modifyAttachments`

Description: Modifies attachments of student submission. Attachments may only be added to student submissions belonging to course work objects with a `workType` of `ASSIGNMENT`. This request must be made by the Developer Console project of the [OAuth client ID](<https://support.google.com/cloud/answer/6158849>) used to create the corresponding course work item. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting user is not permitted to access the requested course or course work, if the user is not permitted to modify attachments on the requested student submission, or for access errors. * `INVALID_ARGUMENT` if the request is malformed. * `NOT_FOUND` if the requested course, course work, or student submission does not exist.

Google API Documentation

Endpoint: `courses.courseWork.studentSubmissions.patch`

HTTP Method: PATCH

Path: `v1/courses/{courseId}/courseWork/{courseWorkId}/studentSubmissions/{id}`

Description: Updates one or more fields of a student submission. See `google.classroom.v1.StudentSubmission` for details of which fields may be updated and who may change them. This request must be made by the Developer Console project of the [OAuth client ID](<https://support.google.com/cloud/answer/6158849>) used to create the corresponding course work item. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting developer project did not create the corresponding course work, if the user is not permitted to make the requested modification to the student submission, or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course, course work, or student submission does not exist.

Google API Documentation

Endpoint: `courses.courseWork.studentSubmissions.turnIn`

HTTP Method: POST

Path: `v1/courses/{courseId}/courseWork/{courseWorkId}/studentSubmissions/{id}:turnIn`

Description: Turns in a student submission. Turning in a student submission transfers ownership of attached Drive files to the teacher and may also update the submission state. This may only be called by the student that owns the specified student submission. This request must be made by the Developer Console project of the [OAuth client ID](https://support.google.com/cloud/answer/6158849) used to create the corresponding course work item. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course or course work, turn in the requested student submission, or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course, course work, or student submission does not exist.

Google API Documentation

Endpoint: `courses.courseWork.studentSubmissions.return`

HTTP Method: POST

Path: `v1/courses/{courseId}/courseWork/{courseWorkId}/studentSubmissions/{id}:return`

Description: Returns a student submission. Returning a student submission transfers ownership of attached Drive files to the student and may also update the submission state. Unlike the Classroom application, returning a student submission does not set `assignedGrade` to the `draftGrade` value. Only a teacher of the course that contains the requested student submission may call this method. This request must be made by the Developer Console project of the [OAuth client ID](<https://support.google.com/cloud/answer/6158849>) used to create the corresponding course work item. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course or course work, return the requested student submission, or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course, course work, or student submission does not exist.

Google API Documentation

Endpoint: `courses.courseWork.studentSubmissions.get`

HTTP Method: GET

Path: `v1/courses/{courseId}/courseWork/{courseWorkId}/studentSubmissions/{id}`

Description: Returns a student submission. * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course, course work, or student submission or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course, course work, or student submission does not exist.

Google API Documentation

Endpoint: `courses.students.delete`

HTTP Method: DELETE

Path: `v1/courses/{courseId}/students/{userId}`

Description: Deletes a student of a course. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to delete students of this course or for access errors. * ``NOT_FOUND`` if no student of this course has the requested ID or if the course does not exist.

Google API Documentation

Endpoint: `courses.students.list`

HTTP Method: GET

Path: `v1/courses/{courseId}/students`

Description: Returns a list of students of this course that the requester is permitted to view. This method returns the following error codes: * `NOT_FOUND` if the course does not exist. * `PERMISSION_DENIED` for access errors.

Google API Documentation

Endpoint: `courses.students.create`

HTTP Method: POST

Path: `v1/courses/{courseId}/students`

Description: Adds a user as a student of a course. Domain administrators are permitted to [directly add](https://developers.google.com/classroom/guides/manage-users) users within their domain as students to courses within their domain. Students are permitted to add themselves to a course using an enrollment code. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting user is not permitted to create students in this course or for access errors. * `NOT_FOUND` if the requested course ID does not exist. * `FAILED_PRECONDITION` if the requested user's account is disabled, for the following request errors: * `CourseMemberLimitReached` * `CourseNotModifiable` * `UserGroupsMembershipLimitReached` * `InactiveCourseOwner` * `ALREADY_EXISTS` if the user is already a student or teacher in the course.

Google API Documentation

Endpoint: `courses.students.get`

HTTP Method: GET

Path: `v1/courses/{courseId}/students/{userId}`

Description: Returns a student of a course. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to view students of this course or for access errors. * ``NOT_FOUND`` if no student of this course has the requested ID or if the course does not exist.

Google API Documentation

Endpoint: `courses.topics.list`

HTTP Method: GET

Path: `v1/courses/{courseId}/topics`

Description: Returns the list of topics that the requester is permitted to view. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course does not exist.

Google API Documentation

Endpoint: `courses.topics.patch`

HTTP Method: PATCH

Path: `v1/courses/{courseId}/topics/{id}`

Description: Updates one or more fields of a topic. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting developer project did not create the corresponding topic or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``FAILED_PRECONDITION`` if there exists a topic in the course with the same name. * ``NOT_FOUND`` if the requested course or topic does not exist

Google API Documentation

Endpoint: `courses.topics.create`

HTTP Method: POST

Path: `v1/courses/{courseId}/topics`

Description: Creates a topic. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course, create a topic in the requested course, or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``ALREADY_EXISTS`` if there exists a topic in the course with the same name. * ``NOT_FOUND`` if the requested course does not exist.

Google API Documentation

Endpoint: `courses.topics.get`

HTTP Method: GET

Path: `v1/courses/{courseId}/topics/{id}`

Description: Returns a topic. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course or topic, or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course or topic does not exist.

Google API Documentation

Endpoint: `courses.topics.delete`

HTTP Method: DELETE

Path: `v1/courses/{courseId}/topics/{id}`

Description: Deletes a topic. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not allowed to delete the requested topic or for access errors. * ``FAILED_PRECONDITION`` if the requested topic has already been deleted. * ``NOT_FOUND`` if no course or topic exists with the requested ID.

Google API Documentation

Endpoint: `courses.teachers.delete`

HTTP Method: DELETE

Path: `v1/courses/{courseId}/teachers/{userId}`

Description: Removes the specified teacher from the specified course. This method returns the following error codes: *

- * ``PERMISSION_DENIED`` if the requesting user is not permitted to delete teachers of this course or for access errors.
- * ``NOT_FOUND`` if no teacher of this course has the requested ID or if the course does not exist.
- * ``FAILED_PRECONDITION`` if the requested ID belongs to the primary teacher of this course.
- * ``FAILED_PRECONDITION`` if the requested ID belongs to the owner of the course Drive folder.
- * ``FAILED_PRECONDITION`` if the course no longer has an active owner.

Google API Documentation

Endpoint: `courses.teachers.create`

HTTP Method: POST

Path: `v1/courses/{courseId}/teachers`

Description: Creates a teacher of a course. Domain administrators are permitted to [directly add](https://developers.google.com/classroom/guides/manage-users) users within their domain as teachers to courses within their domain. Non-admin users should send an Invitation instead. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting user is not permitted to create teachers in this course or for access errors. * `NOT_FOUND` if the requested course ID does not exist. * `FAILED_PRECONDITION` if the requested user's account is disabled, for the following request errors: * `CourseMemberLimitReached` * `CourseNotModifiable` * `CourseTeacherLimitReached` * `UserGroupsMembershipLimitReached` * `InactiveCourseOwner` * `ALREADY_EXISTS` if the user is already a teacher or student in the course.

Google API Documentation

Endpoint: `courses.teachers.get`

HTTP Method: GET

Path: `v1/courses/{courseId}/teachers/{userId}`

Description: Returns a teacher of a course. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to view teachers of this course or for access errors. * ``NOT_FOUND`` if no teacher of this course has the requested ID or if the course does not exist.

Google API Documentation

Endpoint: `courses.teachers.list`

HTTP Method: GET

Path: `v1/courses/{courseId}/teachers`

Description: Returns a list of teachers of this course that the requester is permitted to view. This method returns the following error codes: * `NOT_FOUND` if the course does not exist. * `PERMISSION_DENIED` for access errors.

Google API Documentation

Endpoint: `courses.aliases.list`

HTTP Method: GET

Path: `v1/courses/{courseId}/aliases`

Description: Returns a list of aliases for a course. This method returns the following error codes: *
`PERMISSION_DENIED` if the requesting user is not permitted to access the course or for access errors. *
`NOT_FOUND` if the course does not exist.

Google API Documentation

Endpoint: `courses.aliases.delete`

HTTP Method: DELETE

Path: `v1/courses/{courseId}/aliases/{alias}`

Description: Deletes an alias of a course. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to remove the alias or for access errors. * ``NOT_FOUND`` if the alias does not exist. * ``FAILED_PRECONDITION`` if the alias requested does not make sense for the requesting user or course (for example, if a user not in a domain attempts to delete a domain-scoped alias).

Google API Documentation

Endpoint: `courses.aliases.create`

HTTP Method: POST

Path: `v1/courses/{courseId}/aliases`

Description: Creates an alias for a course. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to create the alias or for access errors. * ``NOT_FOUND`` if the course does not exist. * ``ALREADY_EXISTS`` if the alias already exists. * ``FAILED_PRECONDITION`` if the alias requested does not make sense for the requesting user or course (for example, if a user not in a domain attempts to access a domain-scoped alias).

Google API Documentation

Endpoint: `courses.announcements.list`

HTTP Method: GET

Path: `v1/courses/{courseId}/announcements`

Description: Returns a list of announcements that the requester is permitted to view. Course students may only view `PUBLISHED` announcements. Course teachers and domain administrators may view all announcements. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting user is not permitted to access the requested course or for access errors. * `INVALID_ARGUMENT` if the request is malformed. * `NOT_FOUND` if the requested course does not exist.

Google API Documentation

Endpoint: `courses.announcements.get`

HTTP Method: GET

Path: `v1/courses/{courseId}/announcements/{id}`

Description: Returns an announcement. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course or announcement, or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course or announcement does not exist.

Google API Documentation

Endpoint: `courses.announcements.create`

HTTP Method: POST

Path: `v1/courses/{courseId}/announcements`

Description: Creates an announcement. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting user is not permitted to access the requested course, create announcements in the requested course, share a Drive attachment, or for access errors. * `INVALID_ARGUMENT` if the request is malformed. * `NOT_FOUND` if the requested course does not exist. * `FAILED_PRECONDITION` for the following request error: * AttachmentNotVisible

Google API Documentation

Endpoint: `courses.announcements.delete`

HTTP Method: DELETE

Path: `v1/courses/{courseId}/announcements/{id}`

Description: Deletes an announcement. This request must be made by the Developer Console project of the [OAuth client ID](<https://support.google.com/cloud/answer/6158849>) used to create the corresponding announcement item. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting developer project did not create the corresponding announcement, if the requesting user is not permitted to delete the requested course or for access errors. * ``FAILED_PRECONDITION`` if the requested announcement has already been deleted. * ``NOT_FOUND`` if no course exists with the requested ID.

Google API Documentation

Endpoint: `courses.announcements.modifyAssignees`

HTTP Method: POST

Path: `v1/courses/{courseId}/announcements/{id}:modifyAssignees`

Description: Modifies assignee mode and options of an announcement. Only a teacher of the course that contains the announcement may call this method. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course or course work or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course or course work does not exist.

Google API Documentation

Endpoint: `courses.announcements.getAddOnContext`

HTTP Method: GET

Path: `v1/courses/{courseId}/announcements/{itemId}/addOnContext`

Description: Gets metadata for Classroom add-ons in the context of a specific post. To maintain the integrity of its own data and permissions model, an add-on should call this to validate query parameters and the requesting user's role whenever the add-on is opened in an [iframe](<https://developers.google.com/classroom/add-ons/get-started/iframes/iframes-overview>). This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.announcements.patch`

HTTP Method: PATCH

Path: `v1/courses/{courseId}/announcements/{id}`

Description: Updates one or more fields of an announcement. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting developer project did not create the corresponding announcement or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``FAILED_PRECONDITION`` if the requested announcement has already been deleted. * ``NOT_FOUND`` if the requested course or announcement does not exist

Google API Documentation

Endpoint: `courses.announcements.addOnAttachments.patch`

HTTP Method: PATCH

Path: `v1/courses/{courseId}/announcements/{itemId}/addOnAttachments/{attachmentId}`

Description: Updates an add-on attachment. Requires the add-on to have been the original creator of the attachment. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.announcements.addOnAttachments.list`

HTTP Method: GET

Path: `v1/courses/{courseId}/announcements/{itemId}/addOnAttachments`

Description: Returns all attachments created by an add-on under the post. Requires the add-on to have active attachments on the post or have permission to create new attachments on the post. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.announcements.addOnAttachments.create`

HTTP Method: POST

Path: `v1/courses/{courseId}/announcements/{itemId}/addOnAttachments`

Description: Creates an add-on attachment under a post. Requires the add-on to have permission to create new attachments on the post. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.announcements.addOnAttachments.delete`

HTTP Method: DELETE

Path: `v1/courses/{courseId}/announcements/{itemId}/addOnAttachments/{attachmentId}`

Description: Deletes an add-on attachment. Requires the add-on to have been the original creator of the attachment. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.announcements.addOnAttachments.get`

HTTP Method: GET

Path: `v1/courses/{courseId}/announcements/{itemId}/addOnAttachments/{attachmentId}`

Description: Returns an add-on attachment. Requires the add-on requesting the attachment to be the original creator of the attachment. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.courseWorkMaterials.patch`

HTTP Method: PATCH

Path: `v1/courses/{courseId}/courseWorkMaterials/{id}`

Description: Updates one or more fields of a course work material. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting developer project for access errors. * `INVALID_ARGUMENT` if the request is malformed. * `FAILED_PRECONDITION` if the requested course work material has already been deleted. * `NOT_FOUND` if the requested course or course work material does not exist

Google API Documentation

Endpoint: `courses.courseWorkMaterials.list`

HTTP Method: GET

Path: `v1/courses/{courseId}/courseWorkMaterials`

Description: Returns a list of course work material that the requester is permitted to view. Course students may only view `PUBLISHED` course work material. Course teachers and domain administrators may view all course work material. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting user is not permitted to access the requested course or for access errors. * `INVALID_ARGUMENT` if the request is malformed. * `NOT_FOUND` if the requested course does not exist.

Google API Documentation

Endpoint: `courses.courseWorkMaterials.get`

HTTP Method: GET

Path: `v1/courses/{courseId}/courseWorkMaterials/{id}`

Description: Returns a course work material. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course or course work material, or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if the requested course or course work material does not exist.

Google API Documentation

Endpoint: `courses.courseWorkMaterials.delete`

HTTP Method: DELETE

Path: `v1/courses/{courseId}/courseWorkMaterials/{id}`

Description: Deletes a course work material. This request must be made by the Developer Console project of the [OAuth client ID](<https://support.google.com/cloud/answer/6158849>) used to create the corresponding course work material item. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting developer project did not create the corresponding course work material, if the requesting user is not permitted to delete the requested course or for access errors. * `FAILED_PRECONDITION` if the requested course work material has already been deleted. * `NOT_FOUND` if no course exists with the requested ID.

Google API Documentation

Endpoint: `courses.courseWorkMaterials.getAddOnContext`

HTTP Method: GET

Path: `v1/courses/{courseId}/courseWorkMaterials/{itemId}/addOnContext`

Description: Gets metadata for Classroom add-ons in the context of a specific post. To maintain the integrity of its own data and permissions model, an add-on should call this to validate query parameters and the requesting user's role whenever the add-on is opened in an [iframe](https://developers.google.com/classroom/add-ons/get-started/iframes/iframes-overview). This method returns the following error codes: * `PERMISSION_DENIED` for access errors. * `INVALID_ARGUMENT` if the request is malformed. * `NOT_FOUND` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.courseWorkMaterials.create`

HTTP Method: POST

Path: `v1/courses/{courseId}/courseWorkMaterials`

Description: Creates a course work material. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access the requested course, create course work material in the requested course, share a Drive attachment, or for access errors. * ``INVALID_ARGUMENT`` if the request is malformed or if more than 20 * materials are provided. * ``NOT_FOUND`` if the requested course does not exist. * ``FAILED_PRECONDITION`` for the following request error: * AttachmentNotVisible

Google API Documentation

Endpoint: `courses.courseWorkMaterials.addOnAttachments.get`

HTTP Method: GET

Path: `v1/courses/{courseId}/courseWorkMaterials/{itemId}/addOnAttachments/{attachmentId}`

Description: Returns an add-on attachment. Requires the add-on requesting the attachment to be the original creator of the attachment. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.courseWorkMaterials.addOnAttachments.patch`

HTTP Method: PATCH

Path: `v1/courses/{courseId}/courseWorkMaterials/{itemId}/addOnAttachments/{attachmentId}`

Description: Updates an add-on attachment. Requires the add-on to have been the original creator of the attachment. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.courseWorkMaterials.addOnAttachments.delete`

HTTP Method: DELETE

Path: `v1/courses/{courseId}/courseWorkMaterials/{itemId}/addOnAttachments/{attachmentId}`

Description: Deletes an add-on attachment. Requires the add-on to have been the original creator of the attachment. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.courseWorkMaterials.addOnAttachments.list`

HTTP Method: GET

Path: `v1/courses/{courseId}/courseWorkMaterials/{itemId}/addOnAttachments`

Description: Returns all attachments created by an add-on under the post. Requires the add-on to have active attachments on the post or have permission to create new attachments on the post. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.courseWorkMaterials.addOnAttachments.create`

HTTP Method: POST

Path: `v1/courses/{courseId}/courseWorkMaterials/{itemId}/addOnAttachments`

Description: Creates an add-on attachment under a post. Requires the add-on to have permission to create new attachments on the post. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.posts.getAddOnContext`

HTTP Method: GET

Path: `v1/courses/{courseId}/posts/{postId}/addOnContext`

Description: Gets metadata for Classroom add-ons in the context of a specific post. To maintain the integrity of its own data and permissions model, an add-on should call this to validate query parameters and the requesting user's role whenever the add-on is opened in an [iframe](https://developers.google.com/classroom/add-ons/get-started/iframes/iframes-overview). This method returns the following error codes: * `PERMISSION_DENIED` for access errors. * `INVALID_ARGUMENT` if the request is malformed. * `NOT_FOUND` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.posts.addOnAttachments.create`

HTTP Method: POST

Path: `v1/courses/{courseId}/posts/{postId}/addOnAttachments`

Description: Creates an add-on attachment under a post. Requires the add-on to have permission to create new attachments on the post. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.posts.addOnAttachments.list`

HTTP Method: GET

Path: `v1/courses/{courseId}/posts/{postId}/addOnAttachments`

Description: Returns all attachments created by an add-on under the post. Requires the add-on to have active attachments on the post or have permission to create new attachments on the post. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.posts.addOnAttachments.get`

HTTP Method: GET

Path: `v1/courses/{courseId}/posts/{postId}/addOnAttachments/{attachmentId}`

Description: Returns an add-on attachment. Requires the add-on requesting the attachment to be the original creator of the attachment. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.posts.addOnAttachments.patch`

HTTP Method: PATCH

Path: `v1/courses/{courseId}/posts/{postId}/addOnAttachments/{attachmentId}`

Description: Updates an add-on attachment. Requires the add-on to have been the original creator of the attachment. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.posts.addOnAttachments.delete`

HTTP Method: DELETE

Path: `v1/courses/{courseId}/posts/{postId}/addOnAttachments/{attachmentId}`

Description: Deletes an add-on attachment. Requires the add-on to have been the original creator of the attachment. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.posts.addOnAttachments.studentSubmissions.get`

HTTP Method: GET

Path: `v1/courses/{courseId}/posts/{postId}/addOnAttachments/{attachmentId}/studentSubmissions/{submissionId}`

Description: Returns a student submission for an add-on attachment. This method returns the following error codes: * ``PERMISSION_DENIED`` for access errors. * ``INVALID_ARGUMENT`` if the request is malformed. * ``NOT_FOUND`` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `courses.posts.addOnAttachments.studentSubmissions.patch`

HTTP Method: PATCH

Path: `v1/courses/{courseId}/posts/{postId}/addOnAttachments/{attachmentId}/studentSubmissions/{submissionId}`

Description: Updates data associated with an add-on attachment submission. Requires the add-on to have been the original creator of the attachment and the attachment to have a positive `max_points` value set. This method returns the following error codes: * `PERMISSION_DENIED` for access errors. * `INVALID_ARGUMENT` if the request is malformed. * `NOT_FOUND` if one of the identified resources does not exist.

Google API Documentation

Endpoint: `registrations.delete`

HTTP Method: DELETE

Path: `v1/registrations/{registrationId}`

Description: Deletes a `Registration`, causing Classroom to stop sending notifications for that `Registration`.

Google API Documentation

Endpoint: registrations.create

HTTP Method: POST

Path: v1/registrations

Description: Creates a `Registration`, causing Classroom to start sending notifications from the provided `feed` to the destination provided in `cloudPubSubTopic`. Returns the created `Registration`. Currently, this will be the same as the argument, but with server-assigned fields such as `expiry_time` and `id` filled in. Note that any value specified for the `expiry_time` or `id` fields will be ignored. While Classroom may validate the `cloudPubSubTopic` and return errors on a best effort basis, it is the caller's responsibility to ensure that it exists and that Classroom has permission to publish to it. This method may return the following error codes: * `PERMISSION_DENIED` if: * the authenticated user does not have permission to receive notifications from the requested field; or * the current user has not granted access to the current Cloud project with the appropriate scope for the requested feed. Note that domain-wide delegation of authority is not currently supported for this purpose. If the request has the appropriate scope, but no grant exists, a Request Errors is returned. * another access error is encountered. * `INVALID_ARGUMENT` if: * no `cloudPubsubTopic` is specified, or the specified `cloudPubsubTopic` is not valid; or * no `feed` is specified, or the specified `feed` is not valid. * `NOT_FOUND` if: * the specified `feed` cannot be located, or the requesting user does not have permission to determine whether or not it exists; or * the specified `cloudPubsubTopic` cannot be located, or Classroom has not been granted permission to publish to it.

Google API Documentation

Endpoint: `userProfiles.get`

HTTP Method: GET

Path: `v1/userProfiles/{userId}`

Description: Returns a user profile. This method returns the following error codes: * ``PERMISSION_DENIED`` if the requesting user is not permitted to access this user profile, if no profile exists with the requested ID, or for access errors.

Google API Documentation

Endpoint: `userProfiles.guardians.delete`

HTTP Method: DELETE

Path: `v1/userProfiles/{studentId}/guardians/{guardianId}`

Description: Deletes a guardian. The guardian will no longer receive guardian notifications and the guardian will no longer be accessible via the API. This method returns the following error codes: * ``PERMISSION_DENIED`` if no user that matches the provided ``student_id`` is visible to the requesting user, if the requesting user is not permitted to manage guardians for the student identified by the ``student_id``, if guardians are not enabled for the domain in question, or for other access errors. * ``INVALID_ARGUMENT`` if a ``student_id`` is specified, but its format cannot be recognized (it is not an email address, nor a ``student_id`` from the API). * ``NOT_FOUND`` if the requesting user is permitted to modify guardians for the requested ``student_id``, but no ``Guardian`` record exists for that student with the provided ``guardian_id``.

Google API Documentation

Endpoint: `userProfiles.guardians.list`

HTTP Method: GET

Path: `v1/userProfiles/{studentId}/guardians`

Description: Returns a list of guardians that the requesting user is permitted to view, restricted to those that match the request. To list guardians for any student that the requesting user may view guardians for, use the literal character ``-`` for the student ID. This method returns the following error codes: * ``PERMISSION_DENIED`` if a ``student_id`` is specified, and the requesting user is not permitted to view guardian information for that student, if ``"-`` is specified as the ``student_id`` and the user is not a domain administrator, if guardians are not enabled for the domain in question, if the ``invited_email_address`` filter is set by a user who is not a domain administrator, or for other access errors. * ``INVALID_ARGUMENT`` if a ``student_id`` is specified, but its format cannot be recognized (it is not an email address, nor a ``student_id`` from the API, nor the literal string ``me``). May also be returned if an invalid ``page_token`` is provided. * ``NOT_FOUND`` if a ``student_id`` is specified, and its format can be recognized, but Classroom has no record of that student.

Google API Documentation

Endpoint: `userProfiles.guardians.get`

HTTP Method: GET

Path: `v1/userProfiles/{studentId}/guardians/{guardianId}`

Description: Returns a specific guardian. This method returns the following error codes: * ``PERMISSION_DENIED`` if no user that matches the provided ``student_id`` is visible to the requesting user, if the requesting user is not permitted to view guardian information for the student identified by the ``student_id``, if guardians are not enabled for the domain in question, or for other access errors. * ``INVALID_ARGUMENT`` if a ``student_id`` is specified, but its format cannot be recognized (it is not an email address, nor a ``student_id`` from the API, nor the literal string ``me``). * ``NOT_FOUND`` if the requesting user is permitted to view guardians for the requested ``student_id``, but no ``Guardian`` record exists for that student that matches the provided ``guardian_id``.

Google API Documentation

Endpoint: `userProfiles.guardianInvitations.list`

HTTP Method: GET

Path: `v1/userProfiles/{studentId}/guardianInvitations`

Description: Returns a list of guardian invitations that the requesting user is permitted to view, filtered by the parameters provided. This method returns the following error codes: * ``PERMISSION_DENIED`` if a ``student_id`` is specified, and the requesting user is not permitted to view guardian invitations for that student, if ``"-`"` is specified as the ``student_id`` and the user is not a domain administrator, if guardians are not enabled for the domain in question, or for other access errors. * ``INVALID_ARGUMENT`` if a ``student_id`` is specified, but its format cannot be recognized (it is not an email address, nor a ``student_id`` from the API, nor the literal string ``me``). May also be returned if an invalid ``page_token`` or ``state`` is provided. * ``NOT_FOUND`` if a ``student_id`` is specified, and its format can be recognized, but Classroom has no record of that student.

Google API Documentation

Endpoint: `userProfiles.guardianInvitations.patch`

HTTP Method: PATCH

Path: `v1/userProfiles/{studentId}/guardianInvitations/{invitationId}`

Description: Modifies a guardian invitation. Currently, the only valid modification is to change the ``state`` from ``PENDING`` to ``COMPLETE``. This has the effect of withdrawing the invitation. This method returns the following error codes: * ``PERMISSION_DENIED`` if the current user does not have permission to manage guardians, if guardians are not enabled for the domain in question or for other access errors. * ``FAILED_PRECONDITION`` if the guardian link is not in the ``PENDING`` state. * ``INVALID_ARGUMENT`` if the format of the student ID provided cannot be recognized (it is not an email address, nor a ``user_id`` from this API), or if the passed ``GuardianInvitation`` has a ``state`` other than ``COMPLETE``, or if it modifies fields other than ``state``. * ``NOT_FOUND`` if the student ID provided is a valid student ID, but Classroom has no record of that student, or if the ``id`` field does not refer to a guardian invitation known to Classroom.

Google API Documentation

Endpoint: `userProfiles.guardianInvitations.get`

HTTP Method: GET

Path: `v1/userProfiles/{studentId}/guardianInvitations/{invitationId}`

Description: Returns a specific guardian invitation. This method returns the following error codes: * `PERMISSION_DENIED` if the requesting user is not permitted to view guardian invitations for the student identified by the `student_id`, if guardians are not enabled for the domain in question, or for other access errors. * `INVALID_ARGUMENT` if a `student_id` is specified, but its format cannot be recognized (it is not an email address, nor a `student_id` from the API, nor the literal string `me`). * `NOT_FOUND` if Classroom cannot find any record of the given student or `invitation_id`. May also be returned if the student exists, but the requesting user does not have access to see that student.

Google API Documentation

Endpoint: `userProfiles.guardianInvitations.create`

HTTP Method: POST

Path: `v1/userProfiles/{studentId}/guardianInvitations`

Description: Creates a guardian invitation, and sends an email to the guardian asking them to confirm that they are the student's guardian. Once the guardian accepts the invitation, their `state` will change to `COMPLETED` and they will start receiving guardian notifications. A `Guardian` resource will also be created to represent the active guardian. The request object must have the `student_id` and `invited_email_address` fields set. Failing to set these fields, or setting any other fields in the request, will result in an error. This method returns the following error codes: * `PERMISSION_DENIED` if the current user does not have permission to manage guardians, if the guardian in question has already rejected too many requests for that student, if guardians are not enabled for the domain in question, or for other access errors. * `RESOURCE_EXHAUSTED` if the student or guardian has exceeded the guardian link limit. * `INVALID_ARGUMENT` if the guardian email address is not valid (for example, if it is too long), or if the format of the student ID provided cannot be recognized (it is not an email address, nor a `user_id` from this API). This error will also be returned if read-only fields are set, or if the `state` field is set to a value other than `PENDING`. * `NOT_FOUND` if the student ID provided is a valid student ID, but Classroom has no record of that student. * `ALREADY_EXISTS` if there is already a pending guardian invitation for the student and `invited_email_address` provided, or if the provided `invited_email_address` matches the Google account of an existing `Guardian` for this user.