Google Classroom Add-ons Test Plan

Last updated: Feb 16, 2024

Overview

This document provides a comprehensive testing plan for Google Classroom add-ons. The battery of tests below are intended to fully evaluate an add-on against the Google Workspace Marketplace (GWM) standards and the Classroom Add-ons Requirements. This is the same test plan that will be used by the GWM review team when you submit your add-on for publication.

Many test cases should be performed in both desktop and mobile contexts. Each test is annotated with the applicable contexts, such as Desktop only ▾. You may therefore need to complete each test procedure multiple times when testing an add-on: on a desktop or laptop computer, on an iOS mobile device, and on an Android mobile device. See the test notes for differences in expectations for each test.

The order of test procedures given below is meant to be a logical progression that reduces backtracking when reviewing an application. There is no requirement or expectation that procedures be followed in any specific order.

Note that tests have Functional/Non-Functional labels:

- Functional ▾ tests directly assess the working condition of the add-on. Failure in this category usually means something is “broken.”
- Non-Functional ▾ tests address the behavior and utility of the add-on. A failure in this category typically means that something “feels wrong,” “is confusing,” or “is difficult.”

Escalations & Communications

If you have questions about this process, please file a ticket in our public IssueTracker.

Preparation

The following items are necessary to fully test a Classroom add-on.

Accounts

You'll need access to several accounts:

- A domain administrator account on a Google Workspace for Education domain.
- Two teacher accounts, such as `teacherA@yourEduDomain.com` and `teacherB@yourEduDomain.com`.
- One student account, such as `student@yourEduDomain.com`.

Please follow the onboarding instructions to prepare a Google Workspace for Education demo domain for the purposes of development and testing.

Courses

You will need at least one Google Classroom course to test an add-on. Ensure that the two teacher and one student accounts have been added to this course.

To create a course:

1. Sign into Google Classroom using one of your teacher accounts.
2. Click the **Plus** button at top right.
3. Choose **Create class** (screenshot).
4. Add any name and other details that you would like and click **Create**.
5. Click the **People** tab.
6. Click the **Plus** buttons to add the teacher and student accounts to this class (screenshot). The teacher and student accounts will then be able to access the class.

Hardware

A Classroom add-on review requires the following devices:

- A desktop or laptop computer.
- An Android 5.0+ smartphone with the Google Classroom app installed.
- An Apple iPhone or iPad tablet running iOS 11+ with the Google Classroom app installed.

All devices must have the Google Chrome browser installed. See the Get the Classroom app Help Center article for the device requirements. Optionally, install your product's native mobile app on compatible devices.

Allowlist the add-on

If you have enabled Individual Install in your add-on's GWM SDK App Configuration, you must add it to the domain's allowlist for it to appear for teachers in GWM and Classroom. Follow these steps to ensure that the add-on is on the domain's allowlist:

- Sign into admin.google.com as a domain administrator.
- Go to *Apps* → *GWM Apps* → *Apps list* in the left-hand menu (screenshot).
- Click **Allowlist App** in the top menu.
- Search for the add-on's name, then click the corresponding entry.
- Ensure that **Allow users to install this app** is selected and click the **Continue** button (screenshot).

A. Licensing

“Licensing” refers to granting the user access to the add-on product. These can be evaluated while installing the add-on.

1. GWM listing messaging

- Context(s): Desktop only
- Test type(s): Non-Functional
- Procedure:
  - In the add-on’s GWM Store listing page, read the Overview and Pricing fields.
- Look for:
  - The **Overview** field contains information about the add-on’s licensing requirements.
  - If the **Pricing** field is not “Free,” then the Overview contains information about pricing or how to get a license. [Requirement 1.3, Required]
- How to fix:
  - For each language in which you provide a GWM Store Listing, add details about your add-on’s licensing requirements in the Detailed Description field.
    - If users must purchase a license before using your add-on, also include contact information and instructions for purchasing a license.

2. In-add-on messaging

- Context(s): Desktop only
- Test type(s): Non-Functional
- Procedure:
  - Sign into Chrome with a Google Workspace for Education teacher account that *has not been granted a license for your product*.
  - Open the add-on. Sign in and provide OAuth consent if prompted.
- Look for:
  - A human-readable message should appear if the user is not licensed to use the add-on. [Requirement 1.1, Required]
  - If a message appears, it explains why the add-on cannot be used and suggests a way to fix the problem. [Requirement 1.2, Required]
- How to fix:
  - Provide messaging in your add-on’s Attachment Discovery iframe if the user does not have a license. The message must clearly explain that the user cannot currently use the add-on and provide information to address the issue.
    - For example, you might instruct the user to contact a salesperson, sign into the partner’s website to authorize the add-on, alert an administrator, etc.

B. General

The items listed here do not belong to a specific testing category, but rather should be observed throughout the testing process.

1. GWM approval guidelines

- Context(s): Both desktop and mobile
- Test type(s): Non-Functional
- Procedure:
  - Compare the add-on against the Areas of Review that apply to all integrations.
- Look for:
  - Compliance with the GWM approval criteria. [Requirement 3.1, Required]
- How to fix:
  - Ensure that your application and its configuration comply with the GWM review criteria for **All** and **Google Classroom add-ons** integrations. Please see our Classroom-specific pre-publication guide for a detailed list of items to check.

2. New tabs or windows

- Context(s): Both desktop and mobile
- Test type(s): Non-Functional
- Procedure:
  - This test only applies if a user can open a new tab or window from an add-on iframe.
  - Perform an action in the add-on that opens a new tab or window.
- Look for:
  - An instruction to return to Classroom when the task is complete, such as text, toast, mouseover, etc. [Requirements 3.2 & 5.4, Required]
- How to fix:
  - We **highly recommend** *keeping the user in the iframe*.
  - If not possible to remain in the iframe, it is acceptable for you to open content in new tabs **as long as**:
    - The tab **opens directly** to the relevant content or page.
    - The user sees a message directing them to return to Classroom when finished with the task that opened in the new tab. The message *must be displayed on the page in the new tab*. The message could come in any readable form, such as a notification, on-page text, toast, or image.

3. Journey uses add-ons endpoints only

- Context(s): Both desktop and mobile
- Test type(s): Non-Functional

- Procedure:
  - Observe while performing other tests.
- Look for:
  - The user should *never* be shown a "Share to Classroom" button or link.
  - A new assignment should *never* be created programmatically. [Requirement 3.3, Required ▾]
- How to fix:
  - Remove or prevent from appearing *all* Share to Classroom buttons and CourseWork logic from your add-on integration. The add-ons and CourseWork APIs are currently separate APIs and journeys. Teachers can only access add-ons while manually creating an assignment. Creating assignments programmatically while the teacher is on an add-ons journey is very likely to cause end-user confusion.
  - **Important:** If your add-on opens new tabs, ensure that the pages opened in the new tab similarly do not present any Share to Classroom or CourseWork features.

4. Horizontal page scrolling

- Context(s): Both desktop and mobile ▾
- Test type(s): Non-Functional ▾
- Procedure:
  - Observe in all add-on views.
- Look for:
  - All iframed content can be viewed without scrolling the page horizontally. [Requirement 3.8, Required ▾]
  - **Note**: it is acceptable for a component *within the iframe* to require scrolling, such as a table with many columns.
- How to fix:
  - Adjust content size in iframes to fit in common mobile and desktop browsers.
  - See our iframe implementation details guide page for information about dimensions.

5. Mobile behavior

- Context(s): Mobile only ▾
- Test type(s): Non-Functional ▾
- Procedure:
  - Repeat this test using both the installed Classroom native mobile app and Classroom web using the Google Chrome mobile browser.
  - Open any Teacher View, Student View, or Student Work Review add-on iframe **on a mobile device**. Observe whether the correct content is displayed.
- Look for:

    - If the add-on *does not* support the mobile device, a user-friendly message appears informing the user. The message also indicates which devices are supported.
    - If the add-on *does* support the mobile device, loading any add-on iframe will open the partner's mobile app (if installed) or the partner's mobile website in a browser. The user must be deep-linked directly to the correct content. [Requirement 8.1, Required ▾]
- How to fix:
    - Include logic in your web app to detect if the user is on a mobile device.
        - Ensure that the content is sized appropriately to the device.
        - If your product *does not* support mobile devices, provide a user-friendly message informing the user and indicating *which devices are supported*.
        - If you provide a native mobile app that supports deep-linking, install your native mobile app. Ensure that clicking on attachment cards in Google Classroom opens the correct content *in the native mobile app*.
        - Ensure that clicking on attachment cards in Google Classroom opens the correct content *in your mobile-friendly website*. Uninstall your product's native mobile app from the device if necessary to test this behavior.

C. Sign-in

“Sign-in” refers to the process of logging into the add-on and authenticating a user.

There are a variety of cases to test here, as there are several ways that a user might organize and utilize their accounts. Please read and follow each procedure carefully.

Before proceeding:

- Install the add-on from GWM; it’s possible that this can only be done as a domain administrator. Ensure that the add-on has been added to the app allowlist by a domain administrator if the add-on permits individual install.
- Ensure that you have access to the testing account credentials.
- Create separate Chrome profiles for each of the testing accounts.

Revoke OAuth consent

Important: *You will need to repeat the full sign-in process several times to complete these tests*. To ensure that we are testing from a fresh state, please *take the following actions **before each procedure***:

- Open https://myaccount.google.com/permissions.
- Look for any entries for the add-on being tested.
- If present, click the entry and choose **Remove Access** (screenshot).

1. Teacher sign-in & branding

- Context(s): Desktop only ▾
- Test type(s): Functional ▾ Non-Functional ▾
- Procedure:
  - Sign into Google Classroom as a teacher.
  - Click the **Create** button in the Classwork tab and choose **Assignment** (screenshot).
  - Select the partner attachment from the **Add-ons** area (screenshot).
  - The Attachment Discovery iframe opens.
- Look for:
  - A prompt to sign in using Google appears on the first view after opening the add-on. [Requirement 2.1, Required ▾]
  - The user sees a **Sign in with Google** button when opening the add-on for the first time. [Requirement 2.7, Required ▾]
  - After completing the sign-in flow, the correct user is signed into the add-on. [Requirement 2.1, Required ▾]
  - The add-on iframe shows which user is currently signed in [Requirement 2.8, Recommended ▾]

- How to fix:
    - Sign-in for the Classroom add-ons should be facilitated by the `login_hint` query parameter. This query parameter is not sent if the user has not previously signed in to your application.
        - Ensure that a Google-branded sign-in button is being displayed if your application does not receive the `login_hint` query parameter.
        - If you receive the `login_hint` query parameter, you must display a Google-branded sign-in button if the `login_hint` query parameter does not match the currently signed in user's ID in Classroom.
    - See our Sign the user in and Handle repeat visits walkthrough guides for a detailed discussion of these query parameters and how to handle them.

2. OAuth scopes

- Context(s): Desktop only
- Test type(s): Functional
- Procedure:
    - Uninstall the add-on as a teacher:
        - Click the **Create** button in the Classwork tab and choose **Assignment** (screenshot).
        - Click the three-dots menu in the add-ons area and choose **Explore add-ons** (screenshot).
        - Click the add-on's GWM entry and choose **Uninstall** (screenshot).
        - Close the assignment creation view and reload the page.

      – OR –

      Uninstall the add-on as an admin:
        - Sign into admin.google.com as a domain administrator.
        - Go to *Apps* → *GWM Apps* → *Apps list* in the left-hand menu (screenshot).
        - Click the add-on entry, then choose **Uninstall App** at left (screenshot).
        - Follow the prompts to complete the uninstallation.
    - Wait 15 minutes for the changes to propagate.
    - Reload any Classroom browser tabs.
    - Install the add-on.
      As a teacher:
        - Click the **Create** button in the Classwork tab and choose **Assignment**.
        - Click the three-dots menu in the add-ons area and choose **Explore add-ons**.
        - Click the add-on's GWM entry and choose **Install**.
        - Continue through the prompts to fully install the add-on, allowing any OAuth permissions.

    – OR –

    As an admin :

    - Sign into admin.google.com as a domain administrator.
    - Go to *Apps → GWM Apps → Apps list* in the left-hand menu (screenshot).
    - Click **Install App** at the top of the screen.
    - Click the add-on's GWM entry and choose **Install**.
    - Continue through the prompts to fully install the add-on, allowing any OAuth permissions.

  - As a teacher, create an assignment as in test C1 above.
- Look for:
  - You should see *only one OAuth consent pop-up window*, which should occur when installing the add-on.
  - The OAuth consent pop-up lists all the permissions your add-on will require.
- How to fix:
  - Ensure that the *exact same scopes* are:
    - Attached to your GPC project in the OAuth consent screen settings.
    - Listed in the Google Workspace Marketplace SDK's App Configuration page.
    - Requested by your web application when authenticating the user.

3. Student first-time sign-in

- Context(s): Both desktop and mobile
- Test type(s): Functional Non-Functional
- Procedure:
  - Create an assignment as in test C1 above.
  - Use the add-on to create an attachment.
  - Assign the assignment to the class.
  - Sign into Google Classroom as a student.
  - Click the assignment in the Stream.
  - Click the add-on attachment card.
  - The Student View iframe opens.
- Look for:
  - For activity-type attachments, a prompt to sign in. [Requirement 2.6, Recommended]
  - If prompted to sign in:
    - The student can sign in using Google SSO. [Requirement 2.5, Required]
    - The student sees a **Sign in with Google** button when opening the add-on for the first time. [Requirement 2.7, Required]
    - After completing the sign-in flow, the correct user is signed into the add-on. [Requirement 2.5, Required]

        - The add-on iframe shows which user is currently signed in [Requirement 2.8, Recommended]
- How to fix:
    - Sign-in for the Classroom add-ons should be facilitated by the `login_hint` query parameter. This query parameter is not sent if the user has not previously signed in to your application.
        - Ensure that a Google-branded sign-in button is being displayed if your application does not receive the `login_hint` query parameter.
        - If you receive the `login_hint` query parameter, you must display a Google-branded sign-in button if the `login_hint` query parameter does not match the currently signed in user's ID in Classroom.
    - See our Sign the user in and Handle repeat visits walkthrough guides for a detailed discussion of these query parameters and how to handle them.
    - You can request a unique identifier for the student's submission as well:
        - When a student launches the `studentViewURI`, call the `courses.posts.getAddOnContext` method to retrieve the `submissionId`.
        - We recommend that you associate the `submissionId` and `attachmentId` with any stored records of the student's work. The `attachmentId` is passed as a query parameter to the Student View, Teacher View, and Student Work Review iframes..
        - Likewise, the same `submissionId` is passed as a query parameter to the `studentWorkReviewURI`. This can then be used to present the student's work in the Student Work Review iframe.

4. Repeat sign-in

- Context(s): Both desktop and mobile
- Test type(s): Functional
- Procedure:
    - Create an assignment as in test C1 above.
    - Use the add-on to create an attachment and assign the assignment to the class.
    - Repeat the previous two steps. You should now have at least two assignments.
    - Switch to a Chrome profile for a student in this course.
    - Sign into Google Classroom.
    - Open and view the add-on attachments in both assignments.
- Look for:
    - The teacher is prompted to sign in *exactly once*. [Requirement 2.2, Required]
    - If the student is prompted to sign in, they are prompted *exactly once*. [Requirement 2.5, Required]

- How to fix:
  - Sign-in for the Classroom add-ons should be facilitated by the `login_hint` query parameter. This query parameter is not sent if the user has not previously signed in to your application.
    - Ensure that a Google-branded sign-in button is being displayed if your application does not receive the `login_hint` query parameter.
    - If the user has previously authorized the add-on, the `login_hint` query parameter will be sent. If the `login_hint` value matches the currently signed in user in Classroom, keep the user signed into the add-on. If the value does not match, prompt the user to sign in.
  - See our Sign the user in and Handle repeat visits walkthrough guides for a detailed discussion of these query parameters and how to handle them.

5. Multiple users signed in using the same tab

- Context(s): Desktop only
- Test type(s): Functional
- Procedure:
  - Create an assignment as in test C1 above.
  - Use the add-on to create an attachment and assign the assignment to the class.
  - Sign out of Classroom.
  - Sign into Classroom as a student *while using the same browser tab.*
  - Sign into the add-on as the student.
  - View or complete the add-on attachment content and turn in the assignment.
  - Sign out of Classroom.
  - If the add-on attachment required a student submission, sign into Classroom as the teacher that created the assignment.
  - Grade the student's work.
- Look for:
  - The current Classroom user is signed into the add-on at all times. [Requirement 2.3, Required]
  - If the attachment requires a student submission, the submission is associated with the correct student in Classroom. [Requirement 2.6, Recommended]
- How to fix:
  - Ensure that the add-on is only automatically signing a user in when the `login_hint` query parameter value matches the currently signed in user's ID.
  - If not, prompt the user to sign in again. This will ensure that a user previously signed into Classroom will not be incorrectly signed into the add-on.

6. Multiple users signed into Classroom

- Context(s): Desktop only

- Test type(s): Functional
- Procedure:
  - Create an assignment as in test C1 above using teacher account A.
  - Use the add-on to create an attachment and assign the assignment to the class.
  - Click on the teacher's Google account icon in the top right corner of the Classroom webpage (screenshot).
  - Click on **Add another account**. A new tab will open in the same browser window.
  - Sign into Classroom as a different teacher with account B on the newly opened tab. Ensure that account B is in the same domain as account A.
  - Create an assignment.
  - Open the add-on from the menu and sign-in, if needed.
  - Try adding an attachment to the assignment.
- Look for:
  - The Attachment Discovery iframe displays the correct user account information at all times. [Requirement 2.8, Recommended]
  - No errors occur when creating the second assignment. [Requirement 2.3, Required]
  - Optionally, a user-friendly message directs users to log out of other accounts in the browser window.
- How to fix:
  - If a user accidentally opens the wrong account in the add-on and tries to attach an attachment, an `InvalidAddOnToken` error message will be returned. If this occurs, we recommend that you present a user-friendly message instructing users to sign out of all accounts and try the action again.

7. User signed into different accounts in the partner's product and in Classroom

- Context(s): Desktop only
- Test type(s): Functional
- Procedure:
  - Visit the partner's website as teacher account A.
  - Sign into the partner's website using Google SSO.
  - Sign-in to Classroom as teacher account B, *a different account from the one that used to sign into the website.*
  - Create an assignment as in test C1 above.
  - Use the add-on to create an attachment and assign the assignment to the class.
- Look for:
  - The Attachment Discovery iframe displays the correct user account information at all times. [Requirement 2.8, Recommended]
  - No errors occur when creating the second assignment. [Requirement 2.3, Required]

    - Optionally, a user-friendly message directs users to log out of other accounts in the browser window.
- How to fix:
    - Sign-in for the Classroom add-ons should be facilitated by the `login_hint` query parameter. This query parameter is not sent if the user has not previously signed in to your application.
        - If you don't receive `login_hint`, prompt the user to sign in.
        - If you receive `login_hint`, compare the value to your currently signed in user. If the value doesn't match, prompt the user to sign in.

8. Pop-up blockers

- Context(s): Desktop only
- Test type(s): Functional
- Procedure:
    - Sign into a teacher account's Chrome profile.
    - Navigate to `chrome://settings/content/popups?search=pop-up` in your browser's address bar.
    - Under **Customized behaviors**, add `https://classroom.google.com` to the **Not allowed to send pop-ups or redirects** group (screenshot).
    - Create an assignment as in test C1 above.
- Look for:
    - The user is able to complete the sign-in flow and is correctly signed into the add-on. [Requirement 2.1, Required]
    - If blocking pop-ups prevents the user from completing the sign-in flow, a clear message and/or instructions are provided.
- How to fix:
    - Present a button that initiates the pop-up. Pop-ups are typically allowed if they are the result of a user action.
    - If the user's browser is completely blocking pop-ups and the sign-in or the consent flow is still blocked, the add-on should display a clear message that explains how to address the issue and continue the sign-in flow.

D. Attachment Discovery iframe

The Attachment Discovery iframe is the view that opens when a teacher clicks an add-on in the assignment creation flow.

1. General attachment creation flow

- Context(s): Desktop only ▾
- Test type(s): Functional ▾
- Procedure:
  - Create an assignment as in test C1 above. Sign into the add-on.
  - Choose one or more pieces of content and attach it to the assignment. Click the **Assign** button when finished.
- Look for:
  - The teacher can browse available content within the Attachment Discovery iframe. [Requirement 4.2, Required ▾]
    - **Note**: This *does not mean that the teacher must have access to all material on the partner's platform within the iframe*. It's acceptable to show only material that the teacher has created or saved to their account. In this case, there should be links and/or instructions for how to make materials available in the add-on.
    - When browsing, the teacher should be able to preview the content or activity before attaching it to the assignment. [Requirement 4.3, Required ▾]
  - Any setup or configuration of the content can be done within the iframe. [Requirement 4.4, Recommended ▾]
  - The teacher has tools to help them find content, such as a search bar, links to categories, or filters. [Requirement 4.5, Recommended ▾]
- How to fix:
  - The Attachment Discovery iframe is the landing page for teachers in your add-on. Ensure that teachers have easy access to your product's content in this view, and can browse the content in a user-friendly way.
  - Teachers should *never* be presented with an empty Attachment Discovery view. Provide links and instructions if teachers must take additional steps for content to appear.
  - The teacher must understand the content they are assigning to their students. Provide a mechanism for teachers to preview content before attaching it to the assignment.

2. Activity attachment creation

- Context(s): Desktop only ▾

- Test type(s): Functional
- Procedure:
    - Follow the procedure for test D1 above. Continue this test *only if* at least one of the attachments is an activity, meaning that it requires a student-generated submission.
    - Open Classroom as a student. Click the assignment in the Stream tab, or click **View assignment** in the Classwork tab (screenshot).
    - Observe the position of the attachment cards on screen.
        - Attachment cards in the right side of the screen under the **Your work** label are *activity-type attachments* (screenshot).
        - Attachment cards elsewhere on the screen are *content-type attachments* (screenshot).
- Look for:
    - The attachment(s) is/are correctly shown as activities. [Requirement 4.1, Required]
- How to fix:
    - Ensure that you are providing a valid URL in the `AddOnAttachment.studentWorkReviewUri` field in `CREATE` requests to the `courses.posts.addOnAttachments` endpoint.

3. Grade passback

- Context(s): Desktop only
- Test type(s): Functional
- Procedure:
    - Follow the procedure for test D1 above. Continue this test *only if* at least one of the attachments sets `maxPoints` (screenshot).
- Look for:
    - The attachments that set `maxPoints` have the **Grade sync** label. [Requirement 3.5, Required]
- How to fix:
    - Ensure that you are setting a non-zero, positive integer value in the `AddOnAttachment.maxPoints` field in `CREATE` requests to the `courses.posts.addOnAttachments` endpoint.

4. Copy course

- Context(s): Desktop only
- Test type(s): Functional
- Procedure:
    - Click the **People** tab and ensure that one student test account (Student A) is enrolled in the class as a student.

        - If the second student test account (Student B) is also enrolled in the course, click the three-dot menu next to their name and choose **Remove**.
    - Follow the procedure for test D1 above.
    - If the attachment is an activity, provide a response as Student A.
    - As a teacher, copy the class:
        - Click on the pull-down menu at the top-left of the Classroom webpage. Click **Home** (screenshot).
        - Find the card for the class in which you created the assignment. Click the card's three-dot menu and choose **Copy** (screenshot).
        - Click **Copy** on the following dialog box.
        - You should now see a new card for the copied class. Click it and go to the **Classwork** tab. You should see the same assignment that you created in the original class (screenshot).
        - Add Student A's account to the new course by clicking the **People** tab, and then the **Plus** button next to Student (labels 5 and 6 in this screenshot).
    - As both teacher and Student A, open the attachment in the new course.
    - If the attachment is an activity, provide a response as Student A. Make this response different from the previous one so as to distinguish them.
    - Add another student account (Student B) to the new course, following the same procedure as above.
    - As Student B, open the attachment in the new course.
    - If the attachment is an activity, provide a response as Student B.
- Look for:
    - The assignment with an add-on attachment appears in both classes.
    - Users in both classes can open the add-on attachment. [Requirements 5.5, 6.3, 7.5, Required ▾]
    - If the attachment is an activity:
        - Ensure that Student B's response is correctly recorded.
        - If your product *does not allow* a student to complete the same activity twice, ensure that Student A cannot provide a new, unique response to the activity in the copied class.
        - If your product allows a student to complete the same activity twice, ensure that both of Student A's responses are recorded. [Requirements 5.6, 7.6, Recommended ▾]
- How to fix:
    - Copying a course makes new copies of the original assignments and attachments. *The new copies of add-on attachments will have new `attachmentIds`, and copied assignments will have new `courseId` and `postId` values.*
    - If a user attempts to open an attachment whose `attachmentId` is not in your records, issue a `courses.posts.addOnAttachments.get request` and look at the attachment's `copyHistory`. The `copyHistory` contains mappings

    to the original attachment's `courseId`, `postId`, and `attachmentId`. Use this information to look up the original attachment in your records.
  - If the attachment is an activity, ensure that you are using `(attachmentId, submissionId)` as a composite key for any records of student work.

5. Assign to multiple courses

- Context(s): Desktop only
- Test type(s): Functional
- Procedure:
  - You will need two classes (Class A and Class B) to complete this test. Ensure that your Student A test account is a student in both classes, and that Student B is a student in Class B only.
    - Consider completing test D4 above before proceeding, during which you create a new class with the correct arrangement.
  - Follow the procedure for test D1 above in Class A, but *do not click the **Assign** button yet*.
  - On the right-hand side of the assignment creation view, click the drop-down list under the **For** label (screenshot).
  - Click the checkbox next to Class A and Class B from the drop-down list.
  - Click the **Assign** button.
  - As both teacher and Student A, open the attachment in Class A.
  - If the attachment is an activity, provide a response as Student A in Class A.
  - Click on the pull-down menu at the top-left of the Classroom webpage. Click **Classes** (screenshot).
  - Click the card for Class B.
  - As Student A, open the attachment in Class B.
  - If the attachment is an activity, provide a response as Student A. Make this response different from the previous one so as to distinguish them.
  - As Student B, open the attachment in Class B.
  - If the attachment is an activity, provide a response as Student B.
- Look for:
  - The assignment with an add-on attachment appears in both classes.
  - Users in both classes can open the add-on attachment. [Requirements 5.5, 6.3, 7.5, Required]
  - If the attachment is an activity:
    - Ensure that Student B's response is correctly recorded.
    - If your product *does not allow* a student to complete the same activity twice, ensure that Student A cannot provide a new, unique response to the activity in Class B.

        - If your product allows a student to complete the same activity twice, ensure that both of Student A's responses are recorded. [Requirements 5.6, 7.6, Recommended ▾]
- How to fix:
    - Assigning to multiple courses creates separate copies of the original assignments and attachments. *Each copy of add-on attachments will have different `attachmentIds`, and each assignment will have different `courseId` and `postId` values*.
    - If a user attempts to open an attachment whose `attachmentId` is not in your records, issue a `courses.posts.addOnAttachments.get request` and look at the attachment's `copyHistory`. The `copyHistory` contains mappings to the original attachment's `courseId`, `postId`, and `attachmentId`. Use this information to look up the original attachment in your records.
    - If the attachment is an activity, ensure that you are using `(attachmentId, submissionId)` as a composite key for any records of student work.

E. Teacher View iframe

The Teacher View iframe is the view that opens when a teacher clicks on an add-on attachment card outside of the assignment creation flow.

1. Teacher View behavior

- Context(s): Both desktop and mobile
- Test type(s): Functional
- Procedure:
    - As a teacher, create an assignment with an add-on attachment (per test D1 above).
    - Expand the assignment in the Classwork tab. Click the add-on attachment card in the assignment (screenshot).
- Look for:
    - A preview of the content or activity as students will see it. [Requirement 6.1, Required]
    - Optionally, additional tools or information appropriate to the teacher. These might include answer keys, notes, annotations, editing controls, a snapshot of completion status, etc. [Requirement 6.2, Optional]
- How to fix:
    - The Teacher View iframe loads the URL in the `AddOnAttachment.teacherViewUri` field that was specified when the attachment was created. At minimum, this page *must* show a preview of the attachment content as students will see it.
        - This does not need to be an exact reproduction of the student workflow, but it should allow the teacher to fully explore the material that has been assigned to the students.
        - You can include other information here, such as the items mentioned above. This would be an appropriate place to offer teachers editing tools, allowing them to adjust an attachment without recreating it.

F. Student View iframe

The Student View iframe is the view that opens when a student clicks an add-on attachment card. This is the only flow in which students interact with the add-on.

1. Student View behavior

- Context(s): Both desktop and mobile
- Test type(s): Functional Non-Functional
- Procedure:
  - As a teacher, create an assignment with an activity-type attachment (per test D1 above).
  - As a student in the course, select the assignment created in the previous step in the Classwork tab and click the attachment card (screenshot).
  - Interact with the content and/or complete the activity presented.
- Look for:
  - A new view opens showing the Student View iframe (screenshot).
  - The student can sign in using any of the procedures described above.
  - After signing in, the student can either:
    - View the assigned content or complete the assigned activity in the iframe. [Requirement 5.1, Recommended]

      – OR –

    - Follow a link to open the assigned content or activity in a new tab/window. The link should provide a preview of the content that will open, such as a thumbnail image or title. [Requirement 5.3, Required]
    - If the content or activity opens in a new tab/window, there is clear messaging instructing the student to return to Classroom. [Requirement 5.4, Recommended]
  - If the attachment is an activity, there is *never* a partner-provided button or link that is labeled **Turn in**. [Requirement 5.2, Required]
- How to fix:
  - The Student View iframe loads the URL in the `AddOnAttachment.studentViewUri` field that was specified when the attachment was created. This page should provide the content or activity to the student user.
    - Optionally, *but not recommended*, the Student View iframe can show a link to the content on an external site. This link must give a preview of the linked content, such as a thumbnail or descriptive title.
  - Your application must *never* show a button or link labeled **Turn In**. This wording is reserved for Google Classroom, and is the process by which students finalize an

assignment. A button with this label appears in the top-right corner of the Classroom UI elements for activity-type attachments (screenshot).

G. Student Work Review iframe

The Student Work Review iframe is the view that opens when the teacher views a student submission in the Google Classroom grading UI.

Note: This iframe is *only applicable to add-ons that create activities*. You can skip this section if the add-on provides content-type attachments only.

1. Student Work Review behavior

- Context(s): Both desktop and mobile
- Test type(s): Functional
- Procedure:
    - As a teacher, create an assignment with an activity-type attachment (per test D1 above).
    - As a student, complete the activity and turn in the assignment (per test F1 above).
    - As the teacher, click **View assignment** in the Classwork tab (screenshot).
    - Click the student's card in the Student Work tab (screenshot). A new tab will open.
- Look for:
    - The Student Work Review iframe (screenshot) opens and shows the correct submission for the student that completed the activity. [Requirement 7.1, Required]
        - More information may be present, such as notes, annotations, answer keys, class averages or other statistics, etc. [Requirements 7.3 & 7.4, Optional]
    - Only one student record appears within the iframe. The Classroom navigation tools (screenshot) are the only way to move to another student's submission. [Requirement 7.2, Required]
- How to fix:
    - The Student Work Review iframe loads the URL in the `AddOnAttachment.studentWorkReviewUri` field that was specified when the attachment was created. This field must be provided at creation for activity-type attachments.
    - The loaded `studentWorkReviewUri` must show *exactly one student's work*. Several query parameters are provided when opening this URL that can be used to recall the student's work, including identifiers for the course, assignment, attachment, and individual student submission.
    - When opened, the Student Work Review page must show the student's original work for the teacher to review. Optionally, other information may also be shown, such as described above.

    - *Do not provide any way for the teacher to navigate between students within the iframe*. Teachers will interact with the Classroom UI elements to move between Student Work Review iframes.

2. Grade passback

- Context(s): Both desktop and mobile
- Test type(s): Functional
- Procedure:
    - Follow the procedure for test G1 above. Continue this test *only if* at least one of the attachments sets `maxPoints` (screenshot).
    - Observe whether there is a non-zero value already set in the **Grade field** (screenshot).
    - Wait 30 seconds after loading the student submission in the Student Work Review iframe.
- Look for:
    - A non-zero integer value in the Grade field in the right-hand sidebar.
        - A grade may already be present when you first open the student's submission. If so, the add-on likely sets the grade using stored teacher credentials. [Requirement 3.6, Recommended]

          – OR –

        - If the grade value changes shortly after opening the student's submission, the add-on sets the grade when the teacher views the student's work. [Requirement 3.7, Optional]
- How to fix:
    - If you wish to automatically populate grades in Classroom, the `AddOnAttachment.maxPoints` field must be specified when the attachment is created. `maxPoints` must be a positive integer value, and is the denominator of the attachment grade.
    - If you set `maxPoints` when creating an attachment, your add-on must later issue a `PATCH` request to the `courses.posts.addOnAttachments.studentSubmissions` endpoint. You must specify an integer `pointsEarned` value in this request, which is the numerator of this student's grade for this attachment.

      Note that calls to this endpoint require teacher scopes; you therefore must use a teacher's credentials to make the request. This can be accomplished in two ways:
        - **Option 1**: Make the request using the currently signed in teacher when the `studentWorkReviewUri` is opened.

- **Option 2**: Store the authentication credentials of the teacher that created the attachment. Use these credentials to make the request when the student completes the activity.

Option 1 is easier to implement, but has negative user experience impacts as there is a delay before the grade appears in the Classroom UI and teachers must view each student's submission to populate their gradebook. We strongly recommend that you adopt Option 2 if feasible, as it avoids both problems. Please see our Attachment grades and grade passback walkthrough guide for detailed information about this implementation decision.

Appendix

Glossary

Term	Definition
iframe	An “inline frame”. An HTML page loaded inside a “parent” HTML page. A user interacts with Classroom add-ons through iframes.
Assignment	A task assigned to students within a class. Assignments can have due dates, grades, and can be assigned to all or select students.
Attachment	A piece of learning content or artifact that belongs to an assignment. There are two distinct types of attachments: “content-type” do not require students to produce their own submission, while “activity-type” do require a student submission.
Submission	A student’s response to an activity-type attachment.