

Handling new purchases and free trials

In this article

Step 1. Initial purchase and webhook event

Step 2. Installation

Step 3. Authorization

Step 4. Provisioning customer accounts

When a customer purchases a paid plan, free trial, or the free version of your GitHub Marketplace app, you'll receive the `marketplace_purchase` event webhook with the `purchased` action, which kicks off the purchasing flow.

Note: This article applies to publishing GitHub Apps in GitHub Marketplace only. For more information about publishing GitHub Actions in GitHub Marketplace, see "[Publishing actions in GitHub Marketplace](#)."

If you offer a GitHub App in GitHub Marketplace, your app must identify users following the OAuth authorization flow. You don't need to set up a separate OAuth app to support this flow. See "[Authenticating with a GitHub App on behalf of a user](#)" for more information.

Step 1. Initial purchase and webhook event

Before a customer purchases your GitHub Marketplace app, they select a [listing plan](#). They also choose whether to purchase the app from their personal account or an organization account.

The customer completes the purchase by clicking **Complete order and begin installation**.

GitHub then sends the `marketplace_purchase` webhook with the `purchased` action to your app.

Read the `effective_date` and `marketplace_purchase` object from the `marketplace_purchase` webhook to determine which plan the customer purchased, when the billing cycle starts, and when the next billing cycle begins.

If your app offers a free trial, read the `marketplace_purchase[on_free_trial]` attribute from the webhook. If the value is `true`, your app will need to track the free trial start date (`effective_date`) and the date the free trial ends (`free_trial_ends_on`). Use the `free_trial_ends_on` date to display the remaining days left in a free trial in your app's UI. You can do this in either a banner or in your [billing UI](#). To learn how to handle cancellations before a free trial ends, see "[Handling plan cancellations](#)." See "[Handling plan changes](#)" to find out how to transition a free trial to a paid plan when a free trial expires.

See "[Webhook events for the GitHub Marketplace API](#)" for an example of the `marketplace_purchase` event payload.

Step 2. Installation [🔗](#)

If your app is a GitHub App, GitHub prompts the customer to select which repositories the app can access when they purchase it. GitHub then installs the app on the account the customer selected and grants access to the selected repositories.

At this point, if you specified a **Setup URL** in your GitHub App settings, GitHub will redirect the customer to that URL. If you do not specify a setup URL, you will not be able to handle purchases of your GitHub App.

Note: The **Setup URL** is described as optional in GitHub App settings, but it is a required field if you want to offer your app in GitHub Marketplace. For more information, see "[About the setup URL](#)."

If your app is an OAuth app, GitHub does not install it anywhere. Instead, GitHub redirects the customer to the **Installation URL** you specified in your [GitHub Marketplace listing](#).

When a customer purchases an OAuth app, GitHub redirects the customer to the URL you choose (either Setup URL or Installation URL) and the URL includes the customer's selected pricing plan as a query parameter: `marketplace_listing_plan_id`.

Step 3. Authorization [🔗](#)

When a customer purchases your app, you must send the customer through the OAuth authorization flow:

- If your app is a GitHub App, begin the authorization flow as soon as GitHub redirects the customer to the **Setup URL**. Follow the steps in "[Authenticating with a GitHub App on behalf of a user](#)."
- If your app is an OAuth app, begin the authorization flow as soon as GitHub redirects the customer to the **Installation URL**. Follow the steps in "[Authorizing OAuth apps](#)."

For either type of app, the first step is to redirect the customer to <https://github.com/login/oauth/authorize>.

After the customer completes the authorization, your app receives an OAuth access token for the customer. You'll need this token for the next step.

Note: When authorizing a customer on a free trial, grant them the same access they would have on the paid plan. You'll move them to the paid plan after the trial period ends.

Step 4. Provisioning customer accounts [🔗](#)

Your app must provision a customer account for all new purchases. Using the access token you received for the customer in [Step 3. Authorization](#), call the "[Apps](#)" endpoint. The response will include the customer's `account` information and show whether they are on a free trial (`on_free_trial`). Use this information to complete setup and provisioning.

Note: In the current version of GitHub Marketplace, it's possible for a customer to purchase your app through GitHub Marketplace when they already have an existing account purchased from your app's website. If you find that you already have an account set up for the customer who purchased your app, please report the "double" purchases to [GitHub Support](#).

If the purchase is for an organization and per-user, you can prompt the customer to

choose which organization members will have access to the purchased app.

You can customize the way that organization members receive access to your app. Here are a few suggestions:

Flat-rate pricing: If the purchase is made for an organization using flat-rate pricing, your app can [get all the organization's members](#) via the API and prompt the organization admin to choose which members will have paid users on the integrator side.

Per-unit pricing: One method of provisioning per-unit seats is to allow users to occupy a seat as they log in to the app. Once the customer hits the seat count threshold, your app can alert the user that they need to upgrade through GitHub Marketplace.

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)