Globality

Webhooks Documentation

Integration Guide for Enterprise Customers

VERSION 7

What Are Webhooks?

A webhook (also called a web callback or HTTP push API) is a way for applications to provide other applications with real-time notifications. Unlike typical APIs where you would need to poll for data very frequently to obtain it in real time, webhooks send notifications as an event happens on the Platform.

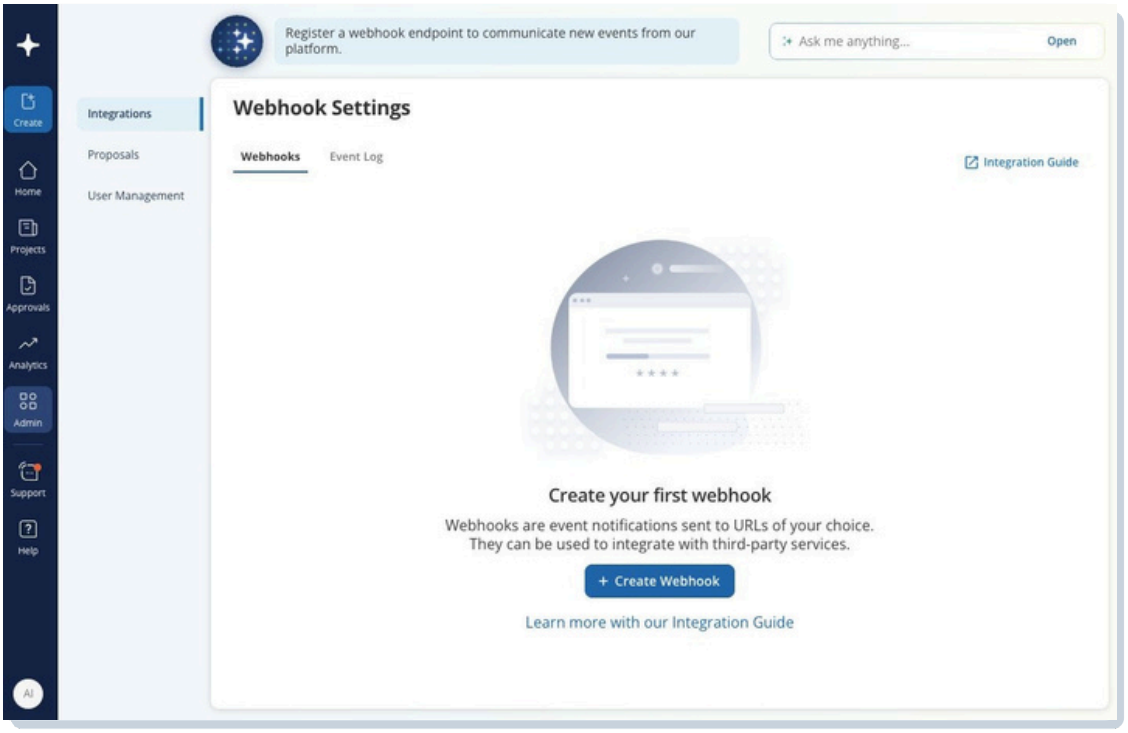
How to Configure Webhooks in Globality

Developer Role Requirements

To access and configure webhooks in Globality, users must be assigned the Developer role. An administrator on your account can assign this role to your user if needed. If an administrator does not see the Developer role available on your company's account, they can reach out to support@globality.com to add this role.

How to Navigate to the Webhooks Settings Page

Once you log in as a user with the Developer role you can navigate to the Webhook Settings page from the main menu on the left. You will be prompted to create your first webhook configuration.



Creating a New Webhook

Next fill out the following information:

- **Endpoint:** URL of your application endpoint where you wish to receive notifications.
- **Description:** A text description of this webhook endpoint.
- **Events to send:** Choose the Globality events to which you would like to subscribe:
  - Brief Completed – triggered when a project brief has been completed.
  - Brief Updated – triggered when there are changes in the brief answers.
  - Proposal Submitted – triggered when a provider submits a proposal.
  - Proposal Approved – triggered when a proposal is approved for award.
  - Award Completed – triggered when a provider's proposal has been approved and awarded for a specific project.
- Saving will finalize your configuration and you can view the newly configured webhook.

Editing Webhook Configurations

You have the ability to edit the webhook details and to take actions such as disable or delete.

- **Update details:** Edit the configuration's endpoint, description, or event subscriptions.
- **Enable / Disable:** Toggles on or off the sending of webhook notifications to the configuration's endpoint. If events happen while the configuration is disabled, they are not queued, and the endpoint will not receive these event notifications when re-enabled.
- **Delete:** Permanently deletes a configuration. Requests will no longer be sent to the configured endpoint.
- **Viewing security token:** You may reveal the security token by clicking the "Show" button. Alternatively, you may copy it directly to your clipboard without showing it.

Event Types

Currently, five event types are supported by webhooks on the Globality Platform. We will continue to add event types over time. If there are specific event types you would like to utilize, please reach out to support@globality.com.

Brief Completed **Brief Updated** **Proposal Submitted** **Proposal Approved** **Award Completed**

Brief Completed

This event is triggered when a project brief is completed in Globality. The payload schema consists of the event type, the ID of the project whose brief has been completed, and the timestamp of when it was completed.

Example Payload


```
{"globalityEvent": "ProjectBriefCompleted",
 "projectId": "e67d7273-6236-4100-af32-9faab3fbddef",
 "timestamp": "2021-12-22T11:52:01-08:00"}
```


Brief Updated

This event is triggered when there are changes in the brief answers for the project. The payload schema consists of the event type, project ID, and the timestamp.

Example Payload


```
{"globalityEvent": "ProjectBriefUpdated",
 "projectId": "eaf79041-5e75-4dc4-beae-b19b7312c34b",
 "timestamp": "2023-02-21T12:34:01-08:00"}
```


Proposal Submitted

This event is triggered when a proposal is submitted by a provider on the platform. The payload schema consists of the event type, project ID, provider ID, and the timestamp.

Example Payload


```
{"globalityEvent": "ProposalSubmitted",
 "projectId": "eaf79041-5e75-4dc4-beae-b19b7312c34b",
 "providerId": "33c0d9fe-a9c3-48c2-88f0-840fe64ea7fd",
 "timestamp": "2023-02-21T12:34:01-08:00"}
```

Proposal Approved

Thiseventis triggered when a proposal is approved and the award process is started. A proposal approval occurs before an award is completed on the platform. The payload schema consists of the event type, project ID, provider ID, and the timestamp.

Example Payload

```
{"globalityevent": "proposalApproved",
 "projectId": "eaf79041-5e75-4dc4-beae-b19b7312c34b",
 "providerId": "33c0d9fe-a9c3-48c2-88f0-840fe64ea7fd",
 "timestamp": "2023-02-21T12:34:01-08:00"}
```

Award Completed

This event is triggered when a project has been awarded to a provider using Globality. The schema for Award Completed consists of globalityEvent, projectId, the timestamp of when it was completed, and the ID of the provider that was awarded.

Example Payload

```
{"globalityEvent": "AwardCompleted",
 "projectId": "e67d7273-6236-4100-af32-9faab3fbddef",
 "providerId": "83b405c4-932a-4859-ae59-b53a90d5be3e",
 "timestamp": "2021-12-22T11:52:01-08:00"}
```

Developing with Globality Webhooks

After setting up a webhook configuration, the registered endpoint can now receive event notifications from the Globality Platform. However, instead of waiting for some event to happen, a testing mechanism is provided for development purposes.

How to Trigger a Test Event

To send a test event from Globality, start by clicking the Send Test button in the Webhooks Settings view. Follow the instructions and select which event you want to trigger, then click Test Webhook. Your request is processed and the contents of the sent payload will appear below in Event Details. A banner will also appear, notifying you of whether it was successfully received by your endpoint.

Viewing Past Events

The event log tab can be used to view previous test events as well as any actual Globality events that have triggered notifications. To differentiate between a real event and a test event, a log entry will have a Test

badge adjacent to the Attempt ID.

Static Security Token Authentication

To authenticate that a request to the configured endpoint is from Globality, check if the verificationToken value matches the configuration's Security Token. The request payload includes the verificationToken within a request header:

Request Header
```
Authorization: Bearer <verificationToken>
```

Responding to a Webhook Request

Webhooks are generally used to respond to an event as soon as possible. With Globality notifications, you can easily extract the project and provider IDs to make subsequent calls to the Globality Project API.

Webhook Retry Logic

If Globality fails to send a request, retries will occur at 1, 3, 6, 12, and 24 hours from failure. After 24 hours, the configuration will automatically be disabled, and an email will be sent notifying you that it has been disabled due to a failure to send an event notification. You will then have the ability to debug the issue.

How to Debug Issues Using the Event Log

The Event Log tab provides not only previous requests but also information for each request sent. The detail option reveals the full event payload information for debugging. Please reach out to support@globality.com if you need any further assistance.

How to Use Webhooks with the Project API

When a request is received for Award Completed, the projectId and providerId from the payload can be used in conjunction with the Project GET endpoint to obtain details about the project:

Project API Endpoint
```
/v2/projects/{projectID}/provider/{providerID}
```

For other endpoints, please refer to the Globality API page in the developer portal.

OAuth 2.0 Authentication

In addition to the static Security Token described above, Globality supports OAuth 2.0 as an enhanced authentication method for webhook delivery. When OAuth 2.0 is enabled on a webhook configuration, Globality will obtain a bearer token from your authorization server before each delivery and include it in the Authorization header of the outbound request. This allows your endpoint to validate incoming requests against your own identity provider rather than a shared static secret.

Prerequisites

Before enabling OAuth 2.0 on a webhook configuration, ensure you have the following information from your authorization server:

- **Token URL:** the token endpoint of your authorization server. Globality will POST to this URL using the client credentials grant type (grant_type=client_credentials) to obtain an access token.
- **Client ID:** The client identifier issued by your authorization server for this integration.
- **Client Secret:** The client secret associated with the Client ID. This is stored encrypted and is never exposed after saving.
- **Scope (optional):** One or more OAuth scopes to request, separated by spaces. Leave blank if your authorization server does not require a scope parameter.

Example Token URL Formats

Identity Provider	Token URL Format
Okta	`https://acme-corp.okta.com/oauth2/v1/token`
Azure AD	`https://login.microsoftonline.com/{tenant-id}/oauth2/v2.0/token`

How to Enable OAuth 2.0 on a Webhook

OAuth 2.0 can be configured when creating a new webhook or when editing an existing configuration. Follow these steps:

1. **Navigate to Webhook Settings**
   Log in as a user with the Developer role and open the Webhook Settings page from the left navigation menu.

2	**Create or edit a webhook** Click Create Webhook for a new configuration, or select Update details on an existing webhook.
3	**Enter the Webhook Endpoint URL** Enter the HTTPS URL of your application endpoint. The endpoint must be publicly accessible and able to accept HTTP POST requests.
4	**Add a Description (optional)** Enter a short description to identify this configuration, such as “Production Okta-authenticated endpoint for sourcing events.”
5	**Select OAuth 2.0 as the authentication type** Open the Authentication Type dropdown and select OAuth 2.0. The form will expand to reveal the four OAuth fields.
6	**Fill in the OAuth 2.0 fields** Enter your Token URL, Client ID, Client Secret, and optionally a Scope. All fields except Scope are required when OAuth 2.0 is selected.
7	**Select events to send** Choose one or more of the available events: Award Completed, Brief Completed, Brief Updated, Proposal Approved, and Proposal Submitted.
8	**Save the configuration** Click Save. Globality will validate connectivity to the Token URL. If the token request fails, an error will be displayed and the configuration will not be saved.

How OAuth 2.0 Authentication Works at Delivery Time

When an event triggers a webhook notification and the configuration is set to OAuth 2.0, Globality performs the following sequence before delivering the payload:

- Globality sends a POST request to the configured Token URL using the client credentials grant type, including the Client ID, Client Secret, and Scope (if provided).
- If a valid access token is returned, Globality includes it in the outbound webhook request as a Bearer token in the Authorization header:

Outbound Header

```
Authorization: Bearer <access_token>
```

- If the token request fails (e.g., invalid credentials, unreachable Token URL), the webhook delivery is treated as a failure and the standard retry logic applies.
- Tokens are not cached between deliveries. A new token is requested for each outbound event.

> Note: Your endpoint should validate the received token against your authorization server to confirm it was issued for this integration. Do not rely solely on the presence of a token as proof of authenticity.

Relationship with the Security Token

When OAuth2.0 is enabled, the static SecurityToken described in the Authentication section is still generated and stored for the configuration. However, it will not be included in the Authorization header of outbound requests while OAuth 2.0 is active. You may still copy or view the Security Token from the configuration detail panel at any time.

Troubleshooting OAuth 2.0 Failures

Token URL unreachable	Verify that the Token URL is publicly accessible from the internet and that your authorization server is running. Check the Event Log for HTTP status codes returned during the token request attempt.
401 Unauthorized from Token URL	The Client ID or Client Secret is incorrect. Update the configuration with the correct credentials and save.
Invalid scope	If your authorization server returns an error indicating an unrecognized scope, clear the Scope field or enter the exact scope value expected by your server.
Endpoint rejects the token	Confirm that your endpoint's token validation logic accepts tokens issued by the same authorization server configured in the webhook. Contact support@globality.com if the issue persists.