DocuBot documentation

Generated by DocuBot

Table of contents

Quick start guide

You'll create and publish a professional, static documentation site from a GitHub repository in under five minutes.

Prerequisites

Before you begin, ensure you have the following:

- A Google account for signing in.
- The URL of a GitHub repository you want to document.
- GitHub authorization if you're connecting a private repository.

1. Sign in to your workspace

Access the DocuBot dashboard by clicking **Sign in** on the homepage. Use your Google account to authenticate. Once signed in, you're redirected to your personal workspace overview.

2. Create a documentation site

Click **Add repository** or **Open workspace** to start a new project. Enter a **Display name** for your project. DocuBot automatically suggests a **URL slug** based on this name, which becomes the public web address for your documentation (e.g., `docubot.cc/your-project`).

3. Connect a repository source

Paste your **GitHub repository URL** into the source field. DocuBot automatically detects the role of the repository, such as a Primary UI or an API reference. If you have a product that spans multiple repositories, you can add additional sources now or later.

4. Select your audiences

Choose the groups your documentation serves, such as **End User**, **Administrator**, or **Developer**. Selecting these audiences automatically generates a preset list of relevant document types, like "Quick start guide" or "API reference," tailored to those users.

5. Launch the documentation pipeline

Review your selections and click **Create documentation site**. This starts the automated generation process. You can monitor the progress via the sync bar on your repository card, which shows the system planning, writing, and publishing your pages.

Verification and next steps

When the sync status changes to **Up to date**, click **View docs** on the repository card to see your live documentation site.

Now that your site is live, you can:

- **Refine document types**: Add or remove specific guides in the repository settings.
- **Change styles**: Select a different general style, such as "Precision Technical," to change the voice of your docs.
- **Configure schedules**: Set your documentation to update daily, weekly, or monthly to stay aligned with your code.

If you need assistance, contact support at support@ademero.com, visit https://www.ademero.com, or call 863-937-0272.

Getting started guide

DocuBot turns your GitHub repositories into structured, static help centers. By connecting your code to our documentation engine, you can provide your users with high-quality, searchable documentation that stays in sync with your latest releases.

In this guide, you'll set up your first documentation site, connect your source code, and publish a live help center.

Access your workspace

To begin managing your documentation projects, you must authenticate with the DocuBot platform.

1. Navigate to the DocuBot dashboard.
2. Click the **Sign in with Google** button.
3. Select your preferred Google account to complete the authentication.

Expected result: You are signed in and redirected to your workspace overview screen.

Create your documentation site

Every documentation site needs a public identity and a unique web address where your users can find it.

1. Click **Add repository** or **Create documentation site** from your dashboard.
2. Enter a **Display name**. This is the internal name you'll see in your workspace.
3. Define your **URL slug**. This creates your public web address at `docubot.cc/your-slug`.
4. Review the live preview link to ensure the address is correct.

Expected result: Your documentation project is initialized with a unique URL.

Connect your GitHub repositories

DocuBot analyzes your source code to generate accurate content. You can link one or multiple repositories to a single documentation site.

1. Enter the **GitHub repository URL** for your product.
2. If your repository is private, click **Connect GitHub** to authorize access via OAuth.
3. Observe the **Role detection**. DocuBot automatically identifies if a repo is a Primary UI, an API reference, or a supporting project. You can manually override these roles if needed.
4. Click **Add another repo** if your product spans multiple repositories.

Expected result: Your source code is linked, and DocuBot is ready to analyze the content.

Define audiences and documentation style

Tailor your documentation to the people who use your product and choose a tone that matches your brand.

1. Select your **Audiences** (such as End User, Administrator, or Developer).
2. Review the **Document types** checklist. DocuBot auto-selects relevant templates based on your audiences, but you can add or remove specific guides.
3. Choose a **General style**. For example, select **Precision Technical** for exact, scan-friendly docs or **Approachable Guided** for a warmer, onboarding-focused tone.

Expected result: A customized set of document types is prepared for generation.

Configure your update schedule

Automate the synchronization between your code changes and your documentation to ensure your help center never falls behind.

1. Select a **Sync frequency** (Daily, Weekly, or Monthly).
2. Choose a specific **Update time** in your local time zone.
3. Verify that your **Preferred time zone** is set correctly in your account settings.

Expected result: An automated update schedule is established for your project.

Verify your live documentation

Once your project is configured, DocuBot begins the initial build process.

1. Monitor the **Sync status indicator** on your repository card.
2. Wait for the status to change to **Up to date**.
3. Click the **View docs** button to open your live site.
4. Check the navigation sidebar to ensure all selected document types are present.

Expected result: Your documentation site is live and accessible at your unique URL.

Next steps

Now that your baseline site is running, you can further refine your content by providing specific scenario focuses or known issues in the **Advanced doc inputs** section of your repository settings.

If you encounter any issues during setup, contact our support team at support@ademero.com, visit https://www.ademero.com, or call 863-937-0272.

Onboarding walkthrough

This guide walks you through creating your first documentation site, connecting a repository, and publishing your first set of static help pages. You can complete this entire process in under ten minutes.

Access your workspace

Sign in to the platform to begin managing your documentation projects.

1. Navigate to the DocuBot homepage.
2. Click the **Sign in** button.
3. Authenticate using your Google account.

Expected result: You are redirected to the DocuBot Overview dashboard where you can see your active repositories and workspace status.

Create a documentation site

Define the public identity and URL for your documentation project.

1. Navigate to the **Add repository** section from the sidebar.
2. Enter a **Display name**. This is the internal name used to identify the project in your workspace.
3. Enter a **URL slug**. This creates the unique web address where your documentation will be hosted.

Expected result: A new documentation project is initialized with a unique URL slug, visible in the live URL preview.

Connect your GitHub repository

Link the source code repository that DocuBot analyzes to generate your content.

1. Enter your **GitHub repository URL** in the source field.
2. If the repository is private, click **Connect GitHub** to authorize access via OAuth.
3. Assign a role to the repository, such as **Primary UI** for your main application or **API reference** for technical specifications.

Expected result: The repository is validated. If it is private, the connection status updates to show it is authorized.

Select your audiences and style

Configure who the documentation is for and the tone it should use.

1. Select your target **Audiences** (such as End User, Administrator, or Developer).
2. Review the **Document types** list. DocuBot automatically selects a preset of templates based on your chosen audiences.
3. Choose a **General style** card to set the overall voice and structural rules for your pages.

Expected result: The system prepares a customized plan for your documentation set based on your selections.

Configure the update schedule

Set how often your documentation stays in sync with your code changes.

1. Select an **Update frequency** (Daily, Weekly, or Monthly).
2. Choose a specific **Update time** using the time picker.
3. Verify that the displayed **Time zone** matches your local preference.

Expected result: A recurring synchronization task is established to ensure your docs never fall behind your code.

Publish and view your documentation

Trigger the initial generation process and view the live site.

1. Click **Create documentation site** to save your settings and start the first sync.
2. Monitor the **Syncing status indicator** on your repository card.
3. Once complete, click the **View docs** button to see your live site.

Expected result: Your documentation site is live at your unique slug, and a **Download PDF** link is available for offline use.

Next steps

Now that your site is live, you can further refine your content:

- How to configure core scenarios
- How to manage support contacts
- How to set up API references

If you need assistance during setup, contact our support team at support@ademero.com, visit https://www.ademero.com, or call 863-937-0272.

First-time setup guide

You'll set up your first DocuBot project to turn your GitHub repositories into a structured, searchable help center. This process establishes your workspace, connects your code, and defines how your documentation looks and feels.

Prerequisites

Before you begin, ensure you have the following:

- A Google account for authentication.
- The URL of at least one GitHub repository you want to document.
- GitHub authorization if you intend to document private repositories.

1. Sign in to your workspace

Establish your user account to access the management dashboard.

1. Navigate to the DocuBot home page.
2. Click **Sign in** or **Open workspace**.
3. Select your Google account to authenticate.

Notice that you are redirected to the dashboard overview screen, which serves as the central hub for all your documentation projects.

2. Create your documentation site

Define the identity and public web address of your help center.

1. Click **Add repository** or **Create documentation site**.
2. Enter a **Display name**. This is the internal name used to identify the project in your workspace.
3. Enter a **URL slug**. This creates the unique public address for your documentation (e.g., `docubot.cc/your-slug`).

Notice that the system provides a live preview of your documentation URL as you type.

3. Connect your GitHub repositories

Link the source code repositories that DocuBot analyzes to generate content.

1. In the **Repository sources** section, enter your GitHub repository URL.
2. If the repository is private, click **Connect GitHub** to authorize access via OAuth.
3. Choose a **Role** for the repository. You can leave this as **Auto detect** or manually specify if it is a Primary UI, a secondary app, or an API reference.

Notice that you can add multiple repositories if your product spans several codebases.

4. Define your target audiences

Select the user groups you serve to automatically determine the most relevant document types.

1. Check the boxes for your intended audiences: **End User**, **Power User**, **Administrator**, or **Developer**.
2. Review the **Document types** list that appears.

Notice that DocuBot pre-selects a default set of templates—like Quick Start Guides or API References—based on your audience choices. You can manually add or remove specific types from this list.

5. Choose your documentation style

Select the tone and structural rules applied to every generated page.

1. Scroll to the **General style** section.
2. Select a style card that matches your brand voice, such as **Precision Technical** for developers or **Approachable Guided** for non-technical users.

Notice that each style includes a summary of its tone and focus to help you choose the best fit.

6. Configure the update schedule

Set the frequency and timing for automated synchronization with your latest code.

1. Select a **Sync frequency**: Daily, Weekly, or Monthly.
2. Enter a **Update time** in your local format.
3. Verify that the **Time zone** matches your current location.

Notice that free accounts are limited to monthly syncs, while paid accounts unlock more frequent updates.

Next actions after setup

Once you click **Create documentation site**, the system enqueues your first synchronization job.

- **Monitor progress**: View the repository card on your dashboard to see the real-time sync status.
- **View your docs**: Once the status changes to "Up to date," click **View docs** to see your live help center.
- **Refine output**: Use the **Advanced doc inputs** in your repository settings to provide specific scenarios or known issues for the AI to prioritize.
- **Add support details**: Go to **Settings** to configure your support email and portal URL so they appear correctly in your generated "Contact and support" pages.

If you encounter issues during setup, contact our support team at support@ademero.com, visit https://www.ademero.com, or call 863-937-0272.

Hello world example

This guide helps you achieve your first success with DocuBot by creating a live documentation site from a public GitHub repository in under five minutes.

Prerequisites

Before you begin, ensure you have the following:

- A Google account for authentication.
- A public GitHub repository URL (e.g., `https://github.com/user/repo`).

Create your first documentation site

Follow these steps to generate your first set of documentation.

1. **Sign in to the dashboard** Navigate to the DocuBot workspace and select **Sign in**. Use your Google account to authenticate and access your personal workspace.

2. **Define your site identity** Select **Add repository** or **Create documentation site**. Enter a **Display name** for your project and choose a unique **URL slug**. This slug determines your public documentation address (e.g., `docubot.cc/your-slug`).

3. **Connect your source code** In the **Repository sources** section, paste the URL of your public GitHub repository. DocuBot automatically detects the repository role. You can leave this as “Auto detect” for your first project.

4. **Select your audience** Choose at least one target audience, such as **End User** or **Developer**. This selection automatically populates a recommended list of document types, such as Quick Start Guides and API References.

5. **Publish the site** Review your settings and select **Create documentation site**. The system enqueues an initial synchronization job to pull your code and generate the static HTML pages.

Verification

Once the synchronization status changes to **Succeeded** on your repository card, you can verify the result:

1. Click the **View docs** button on the repository card.
2. The system opens your new documentation site in a new tab.
3. Confirm that the navigation sidebar contains the document types you selected.

Public API availability

A public API or SDK for programmatic integration with DocuBot is not available in the current repository. All documentation management and site generation must be performed through the DocuBot dashboard.

If you require assistance with custom integrations or have questions regarding API access, please contact our support team:

- **Email**: support@ademero.com
- **Support Portal**: https://www.ademero.com
- **Phone**: 863-937-0272

Sandbox and playground

You can use DocuBot to experiment with different documentation styles, audiences, and document types without affecting your live production help center. By creating a separate test site, you can preview how the AI interprets your code and adjust your settings before finalizing your public documentation.

Create a test documentation site

Establish a separate environment for experimentation that does not use your primary product slug.

1. Navigate to your DocuBot dashboard.
2. Click the **Create documentation site** button.
3. Enter a unique, temporary name in the **URL Slug** field (for example, `product-name-test`). This ensures your sandbox remains separate from your intended production URL.

Connect a sample repository source

Provide the documentation engine with code context to generate your preview pages.

1. In the **Repository sources** section, enter the GitHub repository URL you want to test.
2. If the repository is private, ensure you have connected your GitHub account in your profile settings.
3. The system validates the repository and prepares for the initial sync.

Apply different documentation styles

Compare how different styles change the tone and structure of your content.

1. Scroll to the **General style** section.
2. Select a style card, such as **Precision Technical** for exact, structured data or **Approachable Guided** for a warmer, plain-spoken tone.
3. Notice how the style summary descriptions highlight the different focus areas for each ruleset.

Trigger a manual sync for preview

Generate the documentation immediately to see the results of your style and audience choices.

1. Save your repository settings to return to the dashboard.
2. Locate your test repository card and click the **Rebuild documentation** button.
3. Monitor the **Syncing now** progress bar to see the status of each document type.
4. Once the sync is complete, click the **View docs** link to open your generated sandbox site in a new tab.

Review and delete the test site

Verify the output at the temporary URL and remove the sandbox once your testing is complete.

1. Browse through the generated pages to ensure the tone and document types meet your requirements.
2. If you need to make changes, return to the repository settings and repeat the sync process.
3. When you are finished experimenting, click **Edit settings** on the repository card.
4. Open the **Advanced account settings** at the bottom of the page and follow the prompts to delete the test site. This permanently removes the documentation and its associated storage from your workspace.

Support and assistance

If you have questions about setting up a sandbox environment or need help with style presets, contact the support team:

- **Email:** support@ademero.com
- **Web:** https://www.ademero.com
- **Phone:** 863-937-0272

Core scenario how-to

You can transform your GitHub repositories into a structured, search-ready help center in just a few minutes. This guide walks you through establishing your project identity, connecting your source code, and automating the update schedule.

Create your documentation project

Start by establishing the public identity and web address for your documentation.

1. Sign in to the DocuBot dashboard using your Google account.
2. Click **Add repository** to start a new project.
3. Enter a **Display name**. This is the internal name used to identify the project in your workspace.
4. Choose a unique **URL slug**. This defines your public web address at `docubot.cc/[your-slug]`.

Expected Result: Your project dashboard is created, and the system prepares the source configuration screen.

Connect your GitHub repositories

Link the source code that DocuBot will use to generate your documentation.

1. Enter the **GitHub repository URL** for your product.
2. If your repository is private, click **Connect GitHub** to authorize access via OAuth.
3. Assign a **Role** to the repository. Use **Primary UI** for your main application and **API Reference** for repositories containing OpenAPI or Swagger specifications.
4. Click **Add another repo** if your product spans multiple repositories.

Expected Result: Your repositories are successfully attached as active content sources.

Configure audiences and documentation styles

Tailor the generated content to your specific users and your brand's voice.

1. Select your target **Audiences**, such as End User, Administrator, or Developer. DocuBot uses these to auto-select the most relevant document types.
2. Choose a **General style**, such as **Precision Technical** for exactness or **Approachable Guided** for a warmer tone.
3. Review the **Document types** list. You can manually include or exclude specific templates like Quick Start Guides or Troubleshooting Guides.
4. Use the **Advanced doc inputs** section to list specific scenarios or known issues you want the documentation to prioritize.

Expected Result: The system saves your preferences and prepares a customized documentation plan.

Set your synchronization schedule

Automate the process of keeping your documentation aligned with the latest code your team ships.

1. Select a **Sync frequency**. You can choose Daily, Weekly, or Monthly updates depending on your account tier.
2. Choose a specific **Update time** for the sync to run.
3. Confirm your **Time zone** to ensure the schedule aligns with your local operations.
4. Click **Create documentation site** or **Save changes**.

Expected Result: DocuBot establishes a recurring update schedule and queues the initial generation of your documentation site.

Next steps

Once your site is configured, you can monitor the progress of the documentation build directly from your repository card. When the sync is complete, click **View docs** to see your live help center or **Download PDF** to save a portable version of your documentation.

If you encounter issues during setup, contact our support team at support@ademero.com, visit https://www.ademero.com, or call us at 863-937-0272.

Secondary scenario how-to

This guide covers advanced management tasks for your DocuBot workspace, including handling private repository access, refining repository roles, and managing your documentation exports.

Authorize access to private repositories

DocuBot can generate documentation from both public and private GitHub repositories. While public repositories are accessible by default, private repositories require explicit authorization through GitHub OAuth.

1. Navigate to your workspace **Settings** or the **Edit** screen of a specific repository.
2. Locate the **GitHub connection** section.
3. Click **Connect GitHub** to initiate the authorization process.
4. Follow the prompts on GitHub to grant DocuBot read access to your repositories.

Once connected, your GitHub username appears in your settings, and DocuBot can successfully pull code from your private sources during the next sync.

Manually override repository roles

When you add multiple repositories to a single documentation site, DocuBot automatically detects the role of each source to determine how much weight it receives during content generation. If the automatic detection doesn’t align with your product structure, you can manually override it.

1. Open the **Edit** screen for your repository.
2. Find the **Repository sources** section.
3. Change the **Role** dropdown from “Auto detect” to one of the following:
    - **Primary UI**: The main customer-facing application experience.
    - **UI app**: Secondary experiences, such as admin consoles or mobile apps.
    - **API reference**: Sources containing OpenAPI or Swagger specifications.
    - **Reference only**: Supporting libraries or non-UI codebases.
4. Save your changes to apply the new weighting to your next documentation build.

Modify the automated sync schedule

You can control exactly when and how often DocuBot refreshes your documentation to ensure it stays aligned with your release cycle.

1. Navigate to the **Edit** screen for your repository.
2. Locate the **Update frequency** section.
3. Select your preferred cadence: **Daily**, **Weekly**, or **Monthly**.
4. Set the **Update time** using your local 24-hour clock.
5. Verify your **Time zone** in the workspace settings to ensure the sync runs at the intended moment.

Note that free accounts are limited to monthly syncs, while paid tiers unlock more frequent updates.

Download documentation for offline use

Every time DocuBot publishes your documentation site, it also generates a comprehensive PDF version. This file includes an automated table of contents and internal bookmarks for easy navigation.

1. Go to the **Repositories** page in your dashboard.
2. Locate the card for the project you wish to download.
3. Click the **Download PDF** link.

You can also find a download link in the footer of your live documentation site, making it easy for your end users to save a copy for offline reference.

Update support contact details

The contact information you provide in your workspace settings powers the “Contact and support” page in your public documentation. Keeping this updated ensures your users can always reach the right team.

1. Open your workspace **Settings**.
2. Locate the **Support contacts** section.
3. Update your **Support email**, **Support phone**, or **Support portal URL**.
4. Click **Save support contacts**.

If you need immediate assistance with your DocuBot workspace, you can reach our team at support@ademero.com, visit https://www.ademero.com, or call 863-937-0272.

Workflow guide

This guide walks you through the complete process of creating a documentation site, connecting your source code, and configuring automated updates. By the end of this workflow, you'll have a live, static help center hosted at a custom URL.

Access your workspace

Before you can build your documentation, you must authenticate with the platform to access your personal dashboard.

1. Navigate to the DocuBot homepage.
2. Click the **Sign in** button.
3. Authenticate using your Google account.
4. Once signed in, you'll land on the **Overview** page of your workspace.

Create a documentation site

Define the identity and the public web address where your users will access your help content.

1. Click **Add repository** from the sidebar or the Overview page.
2. Enter a **Display name**. This is the internal name used to identify the project in your workspace.
3. Enter a **URL slug**. This defines the public address of your documentation (e.g., `docubot.cc/your-product-name`).
    - *Note:* Slugs must use lowercase letters, numbers, and dashes only.
4. Verify the live preview URL displayed below the slug field.

Connect repository sources

Link your GitHub repositories to provide the source material for the documentation engine.

1. In the **Repository sources** section, enter your GitHub repository URL.
2. If your repository is private, click **Connect GitHub** to authorize access via OAuth.
3. Assign a **Role** to the repository:
    - **Primary UI:** The main customer-facing application experience.
    - **UI app:** Secondary applications like admin consoles or mobile apps.
    - **API reference:** Repositories containing OpenAPI or Swagger specifications.
    - **Reference only:** Supporting repositories without a user interface.
4. Click **Add another repo** if your product spans multiple repositories.

Select audiences and document types

Tailor the generated content to meet the specific needs of your different user groups.

1. Check the boxes for your target **Audiences** (End User, Power User, Administrator, or Developer).
2. Review the **Document types** list. DocuBot automatically selects a preset based on your audiences.
3. Manually add or remove specific document types to refine your doc set.
4. Choose a **General style** (such as DocuBot Default or Precision Technical) to control the overall tone and voice of the writing.

Configure the update schedule

Set the frequency and timing for automated documentation refreshes to ensure your docs stay aligned with your code.

1. Select a **Sync frequency** (Daily, Weekly, or Monthly).
2. Choose a specific **Update time** for the sync to run.
3. Ensure the **Time zone** matches your local preference. You can update this in your workspace **Settings**.
4. (Optional) Provide **Scenario priorities** or **Known issues** in the advanced inputs to give the engine more context for specific guides.

Publish and verify documentation

Trigger the initial generation process and view your live documentation site.

1. Click **Create documentation site** to save your settings and start the first sync.
2. Return to the **Repositories** page to monitor the progress. You'll see a "Syncing now" status indicator and a progress bar.
3. Once the sync is complete, the status changes to "Up to date."
4. Click **View docs** to open your live documentation site in a new tab.
5. Click **Download PDF** if you wish to view or share the offline version of your documentation.

Support

If you encounter issues during setup or need assistance with repository roles, please contact our support team:

- **Email:** support@ademero.com
- **Web:** https://www.ademero.com
- **Phone:** 863-937-0272

Troubleshooting guide

This guide helps you identify and resolve common issues with repository connectivity, documentation synchronization, and site publishing. Follow these steps to return your documentation site to a healthy state.

GitHub connection and access issues

If you can't see your private repositories or receive an "Access Denied" error, your GitHub authorization is likely missing or expired.

1. Navigate to the **Settings** page in your DocuBot workspace.
2. Locate the **GitHub connection** section to verify your status.
3. Click **Connect GitHub** if the status shows "Not connected" or if you need to refresh your permissions.
4. Confirm that the repository visibility settings on GitHub allow DocuBot to read the content.

Synchronization failures

When a repository card displays a "Failed" status, the automated documentation generation process has stopped.

1. Open the **Repositories** tab in your dashboard.
2. Find the affected repository and read the specific error message displayed on the card.
3. Click **Rebuild documentation** to trigger a manual sync and see if the issue resolves.
4. Verify that the repository URL is still valid and that the default branch has not been deleted or renamed.

Missing API references

If your OpenAPI or Swagger specifications do not appear in the documentation sidebar, check your document type and role settings.

1. Click **Edit settings** on the repository card.
2. Scroll to the **Document types** section and ensure **API Reference** is selected.
3. Check the **Repository sources** section and confirm the role is set to **API reference** or **Auto detect**.
4. Ensure your specification files are in a supported format (JSON or YAML) and are located within the connected repository.

URL slug and publishing errors

If your documentation site fails to load at the expected address or you encounter a slug conflict, verify your configuration.

1. Open the repository settings and locate the **URL slug** field.
2. Click the **Live URL preview** link to confirm the exact public web address.
3. Ensure the slug follows the required format: lowercase letters, numbers, and dashes only.
4. If you receive a conflict error, choose a different, unique slug that isn't already in use by another project.

Contacting support

If the steps above don't resolve your issue, reach out to our support team for further assistance.

- **Email:** support@ademero.com
- **Phone:** 863-937-0272
- **Support Portal:** https://www.ademero.com

When contacting support, include your repository slug and any specific error messages displayed in your workspace.

Workaround and KB article

DocuBot maintains this knowledge base to help you navigate temporary limitations or specific behaviors while we work on permanent improvements. This page provides workarounds for known issues to ensure your documentation pipeline remains reliable.

About known issues

This section tracks documented limitations within the DocuBot environment. We provide these workarounds to help you maintain your documentation workflow without interruption. As we release updates and permanent fixes, we remove resolved items from this list.

Current documented issues

No known issues are documented at this time. The system is performing as expected based on the current configuration. We update this section regularly as new limitations are identified or reported.

How to report a problem

If you encounter a bug, a technical limitation, or an unexpected behavior not listed on this page, please contact our support team for assistance. Providing details about your repository configuration and the specific error helps us resolve the issue faster.

You can reach support through the following channels:

- **Email**: support@ademero.com
- **Support Portal**: https://www.ademero.com
- **Phone**: 863-937-0272

When contacting support, please include your workspace name and the URL slug of the affected documentation site.

Integration guide

You can connect DocuBot to your GitHub repositories and OpenAPI specifications to automate the generation of your documentation. This guide walks you through authorizing access, linking your source code, and verifying that your documentation site is live.

Prerequisites and permissions

Before you begin the integration, ensure you have the following:

- **Google account**: Required to sign in to your DocuBot workspace.
- **GitHub permissions**: You must have read access to the repositories you intend to document.
- **OpenAPI files**: If you want to generate API references, ensure your repository contains valid Swagger or OpenAPI files (JSON or YAML).

Authorize GitHub access

To document private repositories or ensure stable connections to public ones, you must link your GitHub account to your workspace.

1. Open your DocuBot dashboard and navigate to **Settings**.
2. Locate the **GitHub connection** section.
3. Click **Connect GitHub**.
4. Follow the GitHub OAuth prompts to authorize DocuBot.
5. Verify that your GitHub username appears in your profile settings once the redirection is complete.

Connect a repository source

Once authorized, you can link specific repositories to your documentation site.

1. Navigate to the **Repositories** tab in your dashboard.
2. Click **Add repository**.
3. Enter the **GitHub repository URL** for your project.
4. Define a unique **URL slug**. This becomes the public web address for your documentation (e.g., `docubot.cc/your-project`).
5. Click **Save** to add the repository to your tracked projects list.

Configure repository roles

DocuBot uses roles to determine how to prioritize information during the AI generation process.

1. In the repository settings, locate your **Repository sources**.
2. Use the **Role** dropdown to assign a specific purpose to each linked repo:
    - **Primary UI**: The main application experience for your users.
    - **UI app**: Secondary experiences like admin consoles or mobile apps.
    - _API reference_*: Sources primarily used for API specifications.
    - **Reference only**: Supporting repositories without a user interface.
3. Save your changes to update the context budget for future syncs.

Set up OpenAPI ingestion

DocuBot automatically detects API specifications within your connected repositories to build interactive documentation.

1. Ensure your OpenAPI or Swagger files are committed to your repository.
2. DocuBot scans for these files during the synchronization process.
3. Once detected, an interactive API reference is rendered using the Scalar viewer.
4. Check your generated site’s sidebar to find the new **API Reference** links.

Verify the integration

After configuring your sources and roles, confirm that the connection is successful.

1. Return to the **Repositories** page.
2. Look for the **Last sync status** indicator on your repository card.
3. If the status shows **Up to date**, click **View docs** to open your live documentation site.
4. Confirm that the content aligns with your latest code and that all API references are accessible.

Troubleshooting connection issues

If you encounter errors during the integration process, check the following common resolutions:

- **Private repository access**: If a repository fails to sync, ensure your GitHub OAuth token is still valid. You can re-authenticate by clicking **Connect GitHub** in your settings again.
- **Invalid URL slug**: If your site is not accessible, verify that your slug contains only lowercase letters, numbers, and dashes.
- **Missing API docs**: If your API reference does not appear, confirm that your OpenAPI files use standard naming conventions (e.g., `openapi.yaml` or `swagger.json`) and are not located in ignored directories.

If you need further assistance, contact our support team at support@ademero.com, visit our support portal, or call us at 863-937-0272.

Next steps

Now that your repositories are connected, you can configure your sync schedule to keep your documentation updated automatically.

Customization guide

You can tailor your documentation's identity, audience, style, and update frequency to fit your product's specific needs. Follow these steps to adjust your documentation site settings and ensure your content remains relevant to your users.

Update site identity

You can modify the display name and URL slug for your documentation project to align with your branding.

1. Navigate to the repository settings in your dashboard.
2. Enter a new name in the **Display name** field. This name identifies the project within your workspace.
3. Review the **URL slug** field. This defines the unique public web address for your documentation.
4. Observe the live preview below the slug field to see how the final URL will appear (e.g., `docubot.cc/your-product-name`).

Refine target audiences

Adjusting your audience selection changes the recommended document types for your project.

1. Locate the **Audiences** section.
2. Select the roles that best represent your readers: **End User**, **Power User**, **Administrator**, or **Developer**.
3. Notice that selecting a new audience automatically updates the suggested document types.
4. Confirm any prompts regarding document type resets if you previously made custom selections.

Select document types

You can manually include or exclude specific guides to create a documentation set that matches your product's complexity.

1. Scroll to the **Document types** list.
2. Review the available templates grouped by quadrant, such as **Tutorials**, **How-to guides**, **Explanations**, and **Reference**.
3. Check the boxes for the specific guides you want to publish.
4. Monitor the selection count indicator to keep track of your total document set.

Change documentation style

Switch the overall voice and tone of your documentation by selecting a general style that fits your brand's personality.

1. Go to the **General style** section.
2. Choose a style option using the radio buttons. For example:
    - **Precision Technical**: Best for exact, scan-friendly documentation for technical teams.
    - **Approachable Guided**: Best for warm, guided documentation for new or non-technical readers.
3. Read the summary descriptions to understand the tone and structural rules each style applies to your content.

Configure the sync schedule

Set the cadence for automated documentation updates to ensure your help center stays aligned with your latest code changes.

1. Find the **Update frequency** options.
2. Select a cadence: **Daily**, **Weekly**, or **Monthly**.
3. Choose a specific **Update time** using the time selector.
4. Verify your **Time zone** setting to ensure the sync runs at the expected hour.
5. If you select **Weekly** or **Monthly**, pick the specific day of the week or month for the update.

Provide advanced context

You can guide the AI generation process by providing specific scenario priorities or known issues.

1. Open the **Advanced doc inputs** details section.
2. In the **Scenario priorities** text area, list the specific user workflows you want the how-to guides to focus on.
3. In the **Known issues & workarounds** text area, list current limitations and their temporary fixes to populate your workaround articles.
4. Enter each item on a new line for better clarity.

Update support contacts

Configure the contact details that appear on your public “Contact and support” reference page.

1. Navigate to your profile or workspace settings.
2. Enter your **Support email** and **Support phone** number.
3. Provide a **Support portal URL** if you use an external help desk or ticketing system.
4. Save your settings to update the published support page automatically.

If you need assistance with customization, contact support at support@ademero.com, call 863-937-0272, or visit https://www.ademero.com.

Configuration guide

You can configure your DocuBot documentation site to align with your product's audience, style, and update requirements. Follow these steps to establish your site identity, connect your code sources, and customize your documentation output.

1. Define site identity and access

You establish the public-facing identity of your documentation project by setting its name and URL.

1. Enter a **Display name**. This is the internal name for the project in your workspace.
2. Set a **URL slug**. This creates your public documentation address at `docubot.cc/your-slug`.
3. Verify the live preview link to ensure the slug is correct.

2. Manage repository sources and roles

You connect the GitHub repositories that contain your product's code. DocuBot uses these sources to learn your product and generate content.

1. Add your GitHub repository URLs.
2. Assign a **Role** to each source. Use **Primary UI** for your main application. Use **UI app** for secondary experiences, **API reference** for repos containing OpenAPI specs, and **Reference only** for supporting code.
3. Ensure your GitHub account is connected if you're using private repositories.

3. Select audiences and document types

You pick who the documentation is for, which determines the default set of pages DocuBot generates.

1. Check the boxes for your target **Audiences**, such as End User, Administrator, Power User, or Developer.
2. Review the **Document types** list. DocuBot auto-selects a preset based on your audiences, but you can manually toggle specific types like "Quick start guide" or "API reference" on or off.

4. Apply a general documentation style

You set the overall voice and structural rules for your entire documentation set.

1. Choose a **General style** card.
2. Select **DocuBot Default** for a balanced, friendly tone.
3. Select **Precision Technical** for exact, scan-friendly developer docs.
4. Select **Approachable Guided** for warm, plain-language onboarding.

5. Configure the synchronization schedule

You automate how often DocuBot refreshes your documentation from your source code.

1. Select a **Sync frequency**. Free accounts are limited to **Monthly** syncs, while paid accounts unlock **Weekly** and **Daily** options.
2. Choose a specific **Update time** and verify your **Time zone**.
3. If using weekly or monthly syncs, select the specific day you want the update to run.

6. Provide advanced scenario and issue context

You can improve the relevance of your guides by providing specific details about your product's workflows and known limitations.

1. Open the **Advanced doc inputs** section.
2. List your most important user workflows in the **Scenario priorities** area. DocuBot uses these to prioritize content in how-to guides.
3. List known bugs or limitations in the **Known issues & workarounds** area to generate helpful KB articles.

7. Set up support contact information

You configure the contact details that appear on your public "Contact and support" page.

1. Navigate to your **Settings** or **Profile**.

2. Enter your **Support email**, **Support phone**, and **Support portal URL**.
3. Save your changes to ensure users can find help when they need it.

Verification and next steps

Once you save your configuration, DocuBot queues an initial sync job. You can monitor the progress on your repository card in the dashboard. When the status changes to **Up to date**, your documentation is live at your configured URL slug.

If you need assistance with your configuration, contact us at support@ademero.com, visit https://www.ademero.com, or call 863-937-0272.

Migration guide

This guide helps you transition your existing documentation workflow from a legacy system to DocuBot’s automated platform.

Audit existing documentation

Identify which of your legacy documents map to DocuBot’s audience-based framework. Categorize your current help articles by target audience, such as End User, Administrator, or Developer. This categorization allows you to select the correct document types during the setup process in the DocuBot dashboard.

Prepare repository sources

Ensure all source code and OpenAPI specifications are available in your GitHub repositories. DocuBot requires access to these sources to generate accurate, code-aligned documentation. Verify your GitHub connection status in the workspace settings to ensure DocuBot can retrieve the necessary context for AI generation.

Configure the DocuBot workspace

Create a new documentation site within your dashboard. You must define a display name and a unique public URL slug. The slug determines the web address where your documentation will live (for example, `docubot.cc/your-product`). Once configured, your workspace is ready for its first synchronization.

Execute initial synchronization

Trigger the first build to generate your new documentation set. The system processes your repository sources and publishes the first version of your static site. You can monitor the progress through the sync status indicator and verify completion by checking the last sync timestamp on your repository card.

Validate generated output

Review the generated pages and PDF exports for accuracy and style consistency. Use the “View docs” button to browse the live site and the PDF export link to check the formatted document. Ensure the content matches your selected general style and that all technical details are correctly represented.

Update public links

Redirect your users from legacy documentation URLs to the new DocuBot site. Use your reserved public URL slug to point users toward the new SEO-friendly routes. If you have integrated API references, ensure your developer links point to the new interactive API reference pages.

Rollback guidance

If you need to delay your migration or revert to your legacy system, you can disable the automated sync schedule in the repository settings. To completely remove the new documentation from public view, you may delete the URL slug associated with the project.

If you encounter issues during your migration, contact support through the following channels:

- **Email**: support@ademero.com
- **Web**: https://www.ademero.com
- **Phone**: 863-937-0272

Installation and setup guide

In this guide, we will set up your DocuBot workspace and publish your first documentation site from a GitHub repository.

Prerequisites

Before we begin, ensure you have the following:

- A valid Google account for authentication.
- The URL of at least one GitHub repository you want to document.
- GitHub account permissions to authorize access to your repositories.

Step 1: Sign in to your workspace

We start by accessing the DocuBot dashboard.

1. Navigate to the DocuBot home page.
2. Click **Sign in** or **Open workspace**.
3. Select your Google account to complete the authentication.

Expected Result: You see the Overview screen of your new DocuBot workspace.

Step 2: Initialize a new documentation site

Now we begin the process of creating a dedicated home for your project's documentation.

1. From the sidebar, click **Add repository**.
2. Alternatively, click the **Manage repositories** card on the Overview page.

Expected Result: The "Create documentation site" form appears.

Step 3: Define your documentation identity

We need to establish how your documentation appears to the public.

1. Enter a **Display name**. This is the internal name used to identify the project in your dashboard.
2. Review the **URL slug**. DocuBot suggests a slug based on your display name, but you can customize it. This slug defines the public web address (e.g., `docubot.cc/your-slug`).

Expected Result: Your project has a unique identity and a reserved public URL.

Step 4: Connect your repository sources

We will now link your source code to the documentation engine.

1. In the **Repository sources** section, paste your GitHub repository URL.
2. If your repository is private, click **Connect GitHub** to authorize DocuBot via OAuth.
3. Notice that DocuBot automatically detects the role of the repository (such as Primary UI or API reference). You can manually override this if necessary.

Expected Result: The repository status updates to show it is successfully connected.

Step 5: Configure audiences and document types

We tailor the content to the people who will read it.

1. Select one or more **Audiences** (End User, Power User, Administrator, or Developer).
2. Observe that DocuBot auto-selects a preset list of **Document types** based on your audiences.
3. Add or remove specific document types to refine exactly what we will publish.

Expected Result: Your documentation set is customized for your target users.

Step 6: Set documentation style and automation

We choose the voice of the documentation and how often it stays in sync with your code.

1. Select a **General style** (such as DocuBot Default or Precision Technical) to control the tone and layout.
2. Choose an **Update frequency** (Daily, Weekly, or Monthly).
3. Set the **Update time** and verify your **Time zone** to ensure syncs happen when you expect them.

Expected Result: A synchronization schedule is active and a visual style is applied.

Step 7: Verify deployment

Finally, we confirm that your documentation is live.

1. Click **Create documentation site**.
2. Wait for the initial sync job to complete. You can monitor progress on the repository card.
3. Once the status shows “Up to date,” click **View docs** to see your live site.
4. Click **Download PDF** to verify that the portable version of your documentation is ready.

Expected Result: Your documentation site is accessible at your unique URL and the PDF export is available for download.

Troubleshooting initial setup

If you encounter issues during setup, check the following:

- **GitHub Connection Errors:** Ensure you have authorized DocuBot to access the specific organization or repository. If a private repo is not appearing, try reconnecting your GitHub account in **Settings**.
- **Slug Collisions:** If a slug is already in use by another user, you must choose a different, unique URL slug.
- **Sync Failures:** Check the repository card for specific error messages. Common causes include empty repositories or invalid OpenAPI specifications.

If you need further assistance, contact our support team at support@ademero.com or visit https://www.ademero.com. You can also reach us by phone at 863-937-0272.

Next steps

Now that your site is live, you can refine your content by providing **Scenario priorities** or **Known issues** in the repository settings to generate even more helpful guides.

Initial configuration guide

After you connect your first repository, you must configure the core settings of your documentation site. This ensures your help center is professional, targeted to the right readers, and updates automatically as your code changes.

Core settings checklist

Before publishing, verify that you have addressed these critical configuration areas:

- **URL Slug**: The unique public web address for your documentation.
- **Target Audiences**: The specific groups (e.g., End Users, Developers) that define which document types are generated.
- **Documentation Style**: The voice and tone used across all generated pages.
- **Sync Schedule**: The frequency and time for automated updates.
- **Support Contacts**: The contact details displayed to your users for help.

Configuration steps

Follow these steps to complete your initial setup and publish your first documentation set.

1. **Verify your URL slug** Navigate to the repository settings in your dashboard. Locate the **URL slug** field and ensure it correctly represents your product. This slug forms the public address (e.g., `docubot.cc/your-product-name`).
    - **Expected result**: The live URL preview updates to show your locked public web address.
2. **Refine target audiences** In the project configuration, select the audiences that will use your documentation. You can choose from **End User**, **Power User**, **Administrator**, or **Developer**.
    - **Expected result**: DocuBot automatically updates the list of available document types based on your selections.
3. **Select your documentation style** Choose a general style from the available options, such as **DocuBot Default** or **Precision Technical**. Each style provides a different tone and focus for your content.
    - **Expected result**: The selected style becomes active, ensuring a consistent voice across your entire documentation site.
4. **Configure the sync schedule** Set the frequency and local time for DocuBot to pull changes from your GitHub repository.
    - **Note**: Free accounts are limited to **Monthly** syncs. Paid accounts unlock **Weekly** and **Daily** update frequencies.
    - **Expected result**: Your update preferences are saved, and the system displays the next scheduled sync time in your preferred time zone.
5. **Add support contact details** Go to your workspace settings and provide a support email, portal URL, or phone number. These details populate the “Contact and support” page in your generated docs.
    - **Expected result**: Your contact information is saved and ready to be included in the next build.
6. **Trigger the initial build** Return to the repository card and click the **Rebuild documentation** button to start the generation process.
    - **Expected result**: The sync status changes to **Queued** or **Syncing**, and a progress bar appears showing the document generation phases.

Next steps

Once the sync status reaches **Complete**, your documentation is live. You can view the static site by clicking the **View docs** link on your repository card or download a PDF version for offline use.

If you need assistance during configuration, contact support at support@ademero.com, visit https://www.ademero.com, or call 863-937-0272.

First admin user setup

You can complete the initial configuration of your DocuBot workspace to begin generating documentation in just a few minutes. This guide walks you through establishing your identity, linking your code, and scheduling your first documentation sync.

Prerequisites

Before you begin, ensure you have the following:

- A Google account for authentication.
- The URL of at least one GitHub repository you want to document.
- Administrator access to any private repositories you intend to connect.

1. Sign in to the dashboard

Establish your administrator identity using a secure authentication provider.

1. Navigate to the DocuBot landing page.
2. Click the **Sign in** button.
3. Follow the prompts to authenticate with your Google account.

Expected result: You're granted access to the DocuBot workspace management interface and the overview screen.

2. Create your documentation site

Define the public identity and web address for your documentation.

1. Enter a **Display name** for your project. This is the internal name used within your workspace.
2. Configure your **URL slug**. This defines the unique public web address where your documentation will live (e.g., `docubot.cc/your-product`).
3. Review the **Live URL preview** to ensure the address meets your requirements.

Expected result: A unique URL slug is reserved for your product documentation.

3. Connect source repositories

Link the GitHub repositories that contain the source code for your documentation.

1. Enter your **GitHub repository URL**.
2. If the repository is private, follow the prompts to authorize DocuBot via **GitHub OAuth**.
3. Review the **Repository role detection** indicators. DocuBot automatically identifies if a repo is a Primary UI, a secondary app, or an API source.

Expected result: DocuBot identifies the repository roles and prepares to scan for content.

4. Configure audiences and styles

Tailor the generated content to your specific users and brand voice.

1. Select your target **Audiences** (End User, Power User, Administrator, or Developer).
2. Choose a **General style** card, such as "Precision Technical" or "Approachable Guided," to set the tone for the entire doc set.
3. Review the **Document type list**. DocuBot auto-selects a preset of documents based on your chosen audiences, which you can further refine.

Expected result: The system establishes a customized plan for the types of documentation it will generate.

5. Schedule automated updates

Ensure your documentation stays synchronized with your code changes.

1. Select a **Sync frequency** (Daily, Weekly, or Monthly).
2. Use the **Local time picker** to choose when the sync should run.

3. Verify your **Time zone settings** to ensure the schedule aligns with your team's operations.
4. Click **Save** or **Create documentation site**.

Expected result: A recurring synchronization job is established, and your initial documentation generation begins.

Security and access notes

As an administrator, keep the following security contexts in mind:

- **GitHub OAuth permissions:** DocuBot requests specific scopes to read repository content. It does not require write access to your code.
- **Private repositories:** Access to private repos is maintained through your GitHub token. If you revoke access in GitHub, DocuBot will no longer be able to sync those sources.
- **Support configuration:** Ensure your support email and portal URL are configured in your profile settings so they appear correctly in the generated "Contact and support" pages.

Next steps

Once your site is created, you can monitor the sync progress on your repository card. When the status changes to "Up to date," click **View docs** to see your live documentation site.

If you encounter issues during setup, contact our support team at support@ademero.com, visit https://www.ademero.com, or call 863-937-0272.

User management guide

Manage who can access your DocuBot workspace and what they can do. This guide walks you through inviting team members, assigning roles, and revoking access when necessary.

Access workspace settings

To manage your team, you must first locate the administrative controls within your dashboard.

1. Select **Settings** from the primary sidebar.
2. Locate the **Workspace details** or **Team management** section.
3. Verify that you see the list of current active and pending users.

Invite a new team member

You can grant access to your workspace by sending an invitation to a team member's email address.

1. Click the **Invite** button.
2. Enter the email address of the person you want to add.
3. Click **Send invitation**.
4. Notice that the user now appears in your **Pending users** list until they accept the invite via Google Sign-In.

Assign a workspace role

Roles define the level of permission a user has within your workspace. You can set these during the invitation process or update them later.

1. Open the **Role selection** dropdown next to the user's name.
2. Choose between available roles such as **Administrator** or **Editor**.
3. Review the permission level descriptions to ensure the user has the correct access.
4. Save your changes to restrict the user's actions to their assigned role.

Remove or deactivate a user

If a team member no longer requires entry to the workspace, you can revoke their access immediately.

1. Locate the user in the active users list.
2. Click the **Remove user** button.
3. Confirm the revocation in the pop-up window.
4. The user is immediately logged out and can no longer access repositories, sync settings, or documentation configurations.

Verify data impact

Removing a user has specific effects on your workspace data and integrations.

- **Documentation retention**: All documentation generated by the user remains live and accessible at your configured URL slugs.
- **Sync jobs**: Existing sync schedules continue to run as planned.
- **Integrations**: If the removed user provided the GitHub OAuth token for a private repository, that integration may disconnect. You must ensure an active Administrator reconnects their GitHub account to maintain automated syncs.

Understanding workspace roles

Use the following role definitions to determine the appropriate access level for your team.

- **Administrator**: Provides full control over the workspace, including billing, repository setup, sync scheduling, and user management.
- **Editor**: Allows the user to manage repositories and trigger documentation syncs but prevents them from adding or removing other team members.

Support and escalation

If you encounter issues with user seat limits or access permissions, contact the support team for assistance.

- **Email**: support@ademero.com
- **Phone**: 863-937-0272
- **Support Portal**: https://www.ademero.com

Security basics guide

As an administrator, you play a vital role in maintaining the integrity and security of your documentation workspace. DocuBot is designed with a security-first approach, focusing on identity verification, controlled data access, and a static output model that minimizes common web vulnerabilities.

This guide explains the core pillars of the DocuBot security model and provides actionable recommendations to help you keep your documentation environment secure.

DocuBot security model overview

DocuBot handles security through three primary layers: identity management, authorized data ingestion, and static content delivery.

- **Identity and access:** We use Google Sign-In to manage dashboard access. This ensures that only verified users within your organization can manage repository settings, audiences, and sync schedules.
- **Authorized ingestion:** DocuBot only accesses your source code through explicit authorization. For public repositories, we use standard public access. For private repositories, we use GitHub OAuth to establish a secure, time-limited connection.
- **Static delivery:** Unlike traditional documentation platforms that render pages at runtime, DocuBot generates static HTML. This architectural choice significantly reduces the “attack surface” of your documentation site by eliminating the need for a live database or server-side processing during page loads.

Managing repository access

Connecting your source code is the most sensitive part of the documentation process. You have full control over which repositories DocuBot can see and how it accesses them.

Private vs. public repositories

Public repositories are accessible by default. However, if your product lives in a private GitHub repository, you must authorize DocuBot via GitHub OAuth. This process grants DocuBot the specific permissions required to read your code for the purpose of documentation generation.

OAuth token management

When you connect a private repository, DocuBot stores an encrypted access token. You can revoke this authorization at any time through your GitHub account settings or by disconnecting the integration within the DocuBot dashboard. We recommend periodically reviewing your connected integrations to ensure only necessary access is maintained.

Public visibility and URL slugs

Every documentation site you create in DocuBot is assigned a unique URL slug (e.g., `docubot.cc/your-product-name`). It is important to understand the security implications of this public path.

- **Public accessibility:** By default, generated documentation sites are public. This allows your customers and support teams to access help content without a login.
- **Slug naming:** Because slugs are public and unique, avoid using sensitive internal project names or confidential codenames in your URL slug. Use clear, customer-facing product names that align with your public branding.
- **Information disclosure:** DocuBot is designed to generate external-facing documentation. Always review the generated content to ensure that internal-only details, such as private IP addresses or internal server names, are not inadvertently included in your public guides.

Administrative security recommendations

To maintain a healthy security posture, we recommend following these baseline practices:

- **Configure automated syncs:** Set up a regular sync schedule (daily, weekly, or monthly) to ensure your documentation stays aligned with your latest code. This prevents “documentation drift,” where outdated instructions might lead users to perform insecure actions.
- **Maintain support contacts:** Ensure your support email, portal URL, and phone details are accurate in your workspace settings. This ensures that if a user discovers a documentation error or a security concern, they can reach your team immediately.

- **Monitor sync status:** Regularly check the status of your sync jobs in the dashboard. A failed sync or an unexpected change in the document set can be an early indicator of configuration issues or unauthorized repository changes.

Reporting security concerns

We take the security of your documentation seriously. If you identify a vulnerability or have a specific security concern regarding your DocuBot workspace, please contact our security team through the following official channels:

- **Support portal:** https://www.ademero.com
- **Email:** support@ademero.com
- **Phone:** 863-937-0272

Our team will review your report and work with you to resolve the issue promptly.

Backup and recovery guide

You can protect your documentation configurations and restore your site if you encounter configuration errors or data loss. This guide helps you manage your documentation backups and perform a full recovery.

Identify your data scope

Understand where your data lives to ensure you have a complete backup strategy. DocuBot documentation relies on two distinct data sources:

- **Source Content:** Your primary documentation content is stored within your GitHub repositories. As long as your code exists in GitHub, your source content is safe.
- **Site Configuration:** Your workspace settings, including selected audiences, document types, and custom scenario focuses, are managed within the DocuBot dashboard.

Export documentation as PDF

Create a point-in-time backup of your generated documentation to keep a portable record of your help center.

1. Open your repository list in the DocuBot dashboard.
2. Locate the repository you want to back up.
3. Click **Download PDF** on the repository detail card.
4. Save the generated file to a secure local or cloud storage location.

You can also find this export link in the footer of your live documentation site.

Document your site configuration

Keep a record of your specific site settings to make manual restoration easier if a project is accidentally deleted.

1. Navigate to the **Edit settings** page for your repository.
2. Note your selected **Audiences** and **Document types**.
3. Copy any text from the **Scenario priorities** and **Known issues & workarounds** sections in the Advanced doc inputs area.
4. Store these details in a secure internal document.

Restore repository connections

If your documentation stops updating due to a connection failure, you must re-authorize your GitHub access.

1. Go to the **Settings** page in your workspace.
2. Check the **GitHub connection** status.
3. If the status is not “Connected,” click **Connect GitHub**.
4. Follow the authorization flow to restore access to your repositories.

Trigger a recovery rebuild

Once your connections are restored, you can force the system to regenerate your entire documentation site.

1. Return to the **Repositories** page.
2. Locate the project you need to restore.
3. Click **Rebuild documentation**.
4. Monitor the **Syncing now** progress indicator to ensure the process completes.

Verify the restored site

Confirm that your documentation is back online and reflects the latest code.

1. Check the **Last sync** status on your repository card; it should display “Up to date.”
2. Click **View docs** to open your live site.
3. Navigate through a few pages to verify that the content is loading correctly at your public URL slug.

Get help

If you are unable to restore your site or need assistance with a configuration error, contact support:

- **Email:** support@ademero.com
- **Web:** https://www.ademero.com
- **Phone:** 863-937-0272

API quick start

This guide helps you authenticate your environment and make your first request to the DocuBot API. By the end of this tutorial, you'll have a valid connection to the service and understand how to interact with our endpoints.

Prerequisites

Before you begin, ensure you have the following:

- A DocuBot account with the **Developer** audience role assigned.
- Access to the DocuBot dashboard.
- A tool to make HTTP requests, such as `curl`, Postman, or your preferred programming language.

1. Obtain your API credentials

You must have a valid API token to authorize your requests.

1. Sign in to your DocuBot dashboard.
2. Navigate to the **Settings** section.
3. Locate the API key management area and generate a new token.
4. Copy the token and store it securely.

Notice that for security reasons, the full token is only displayed once. If you lose it, you'll need to generate a new one.

2. Configure authentication

DocuBot uses Bearer token authentication for all API interactions. You must include this token in the header of every request you send.

Set up your request headers using the following format:

```
Authorization: Bearer <YOUR_TOKEN>

Content-Type: application/json
```

Replace `<YOUR_TOKEN>` with the string you copied from your dashboard. Keep this token secure and never expose it in client-side code or public repositories.

3. Make your first request

Execute a simple GET request to verify your connectivity. While specific public API entrypoints are managed within your project settings, you can test your connection against your base project URL.

If you are using `curl`, run the following command in your terminal:

```
curl -X GET https://docubot.cc/api/v1/status \

  -H "Authorization: Bearer <YOUR_TOKEN>" \

  -H "Content-Type: application/json"
```

Note: If a public API is not yet documented for your specific repository, please contact our support team at support@ademero.com for assistance with your integration.

4. Verify the response

Confirm that the system recognized your request and returned valid data. A successful request returns a **200 OK** status code.

Check the response body for a JSON object. It typically looks like this:

```json
{
  "status": "success",
  "version": "1.0.0",
  "message": "Connection verified"
}
```

If you encounter a **401 Unauthorized** or **404 Not Found** error, double-check your Bearer token and ensure your account has the correct permissions.

Next steps

Now that you've successfully authenticated, you can explore the full capabilities of the DocuBot API:

- **Interactive API Reference:** Access the full list of available endpoints via the **API Reference** link in your documentation sidebar.
- **Try it feature:** Use the interactive Scalar UI to test requests in real-time directly from your browser.
- **Support:** If you need technical help, visit https://www.ademero.com or call us at 863-937-0272.

Authentication guide

You use two primary methods to authenticate with DocuBot: Google Sign-In for workspace access and GitHub OAuth for repository connections. This guide explains how to set up these connections and manage your permissions.

Sign in with Google

You access the DocuBot dashboard and your personal workspace using your Google account. This is the primary way to manage your documentation sites and sync schedules.

1. Navigate to the DocuBot landing page.
2. Click the **Sign in** button.
3. Select your Google account from the authentication prompt.
4. Verify that you're redirected to the DocuBot overview page.

Once signed in, your profile initials or photo appear in the sidebar to confirm your session is active.

Authorize GitHub access

While public repositories don't require authorization, you must link your GitHub account to DocuBot to sync documentation from private codebases.

1. Open your **Workspace settings** or start the **Add repository** flow.
2. Locate the GitHub connection section and click **Connect GitHub**.
3. Review the requested permissions on the GitHub OAuth authorization screen.
4. Click **Authorize** to establish the secure connection.

DocuBot uses this connection to download the latest state of your private repositories during scheduled syncs.

Verify connection status

You can confirm that your external accounts are correctly linked and active at any time.

1. Go to the **Settings** page in your dashboard.
2. Look for the **GitHub connection** section.
3. Confirm that your GitHub username and the connection timestamp are visible.

You can also see a connection status badge on individual repository detail cards within your workspace.

Manage or revoke permissions

You maintain control over your security by reviewing or removing access to your GitHub account whenever necessary.

1. Navigate to your **User profile** or **Workspace settings**.
2. Find the GitHub connection details.
3. Select the option to **Disconnect GitHub**.
4. Confirm the action to remove DocuBot's access.

Notice that disconnecting your account disables syncing for all private repositories. You'll need to re-authorize access to resume updates for those projects.

Troubleshooting and support

If you encounter issues with Google Sign-In or GitHub authorization, ensure your browser isn't blocking pop-ups or third-party cookies for `docubot.cc`.

For further assistance with authentication or account access, contact our support team:

- **Email**: support@ademero.com
- **Phone**: 863-937-0272

- **Support Portal**: https://www.ademero.com

Information regarding manual rotation of OAuth tokens or session timeout durations is not available in the current documentation.

Common integration how-to

DocuBot transforms your source code into a structured, static help center. By integrating your GitHub repositories, you ensure your documentation stays synchronized with every code change.

Prerequisites

Before you begin, ensure you have the following:

- A DocuBot account (signed in via Google).
- The URL of the GitHub repository you want to document.
- An OpenAPI or Swagger specification file (JSON or YAML) located within your repository if you intend to generate API references.

1. Connect a public repository

The quickest way to start is by linking a public repository. Public repositories do not require additional GitHub authorization.

1. Open your DocuBot workspace and click **Add repository**.
2. Enter a **Display name**. This is the internal name used to identify the project in your dashboard.
3. Enter the **URL slug**. This defines the public web address where your documentation will live (e.g., `docubot.cc/your-product-name`).
4. Paste the **GitHub repository URL** into the source field.
5. Click **Create documentation site**.

Expected result: The repository appears in your **Tracked projects** list, and DocuBot begins the initial scan to identify the codebase structure.

2. Authorize private repository access

To document private codebases, you must grant DocuBot permission to read your GitHub data through a secure OAuth connection.

1. Navigate to **Settings** in the sidebar.
2. Locate the **GitHub connection** section.
3. Click **Connect GitHub**.
4. Follow the prompts on the GitHub authorization screen to grant access.
5. Once redirected, verify that your GitHub username appears as **Connected**.

Expected result: You can now add private repositories as sources using the same steps as public repositories.

3. Link an OpenAPI specification

DocuBot automatically detects API specifications to create interactive, high-performance reference pages.

1. Ensure your repository contains a valid OpenAPI (3.0+) or Swagger (2.0) file.
2. In the repository configuration, ensure the **Developer** audience is selected.
3. Verify that the **API reference** document type is included in your selection.
4. DocuBot will scan the repository, identify the specification, and render it using the built-in Scalar API viewer.

Expected result: An **API reference** entry appears in your documentation sidebar, providing an interactive environment for developers to explore your endpoints.

4. Configure the automated sync schedule

Keep your documentation aligned with your latest releases by setting a recurring update cadence.

1. Go to **Repositories** and click **Edit settings** on your project.
2. Locate the **Update frequency** section.
3. Select your preferred cadence:
    - **Daily**: Updates every 24 hours at a specific time.

    - **Weekly**: Updates on a specific day of the week.
    - **Monthly**: Updates on a specific day of the month.
4. Choose the **Update time** and confirm your **Time zone**.
5. Click **Save changes**.

Expected result: DocuBot automatically triggers a documentation rebuild according to your schedule. You can monitor the status of these updates via the **Last sync** indicator on your project card.

Integration configuration example

When setting up your integration, your configuration typically follows this structure:

Setting	Example Value
Display Name	CloudSync Pro
URL Slug	cloudsync-pro
Primary Source	`https://github.com/your-org/cloudsync-main`
API Source	`https://github.com/your-org/cloudsync-api`
Sync Frequency	Daily
Update Time	03:00 AM

Verification and support

Once your integration is configured, visit your public URL slug to view the live documentation. If the site is still generating, you will see a “Syncing” status in your dashboard.

If you encounter issues connecting a repository or if your API specifications are not being detected, please reach out for assistance:

- **Email**: support@ademero.com
- **Support Portal**: https://www.ademero.com
- **Phone**: 863-937-0272

Webhook guide

You can use webhooks to receive real-time notifications about your documentation lifecycle. This allows you to trigger external workflows, such as updating a status page or notifying a Slack channel, whenever a sync succeeds or fails.

Access webhook settings

To begin the configuration, you must locate the webhook management area within your dashboard.

1. Open the **Workspace settings** menu.
2. Select the **Webhooks** tab.
3. Notice that the Webhooks management screen appears, displaying any existing integrations.

Register a destination URL

DocuBot needs a public endpoint to send event notifications.

1. Enter your public endpoint URL in the **Endpoint URL** field.
2. Save the endpoint.
3. Notice that the system assigns a unique **secret key** to this integration. You will need this key later to verify that incoming requests are authentic.

Select event triggers

You can choose which specific events should trigger a notification to avoid unnecessary traffic to your server.

1. Review the list of available event triggers.
2. Select the events you want to monitor, such as:
    - **Sync Succeeded**: Fired when a documentation update completes successfully.
    - **Sync Failed**: Fired when an error occurs during the generation process.
    - **Initial Generation**: Fired when a new repository is first processed.
3. Save your selection to ensure the webhook only fires for these specific actions.

Verify the payload signature

For security, you should always validate that the requests your server receives actually originate from DocuBot.

1. Locate the **payload signing secret** provided in the dashboard.
2. Configure your server to calculate an **HMAC SHA256** signature of the incoming request body using your secret key.
3. Compare your calculated signature with the one provided in the request header.
4. Reject any requests where the signatures do not match to prevent unauthorized access.

Note: The exact header name for the security signature and the sample JSON payload schema are not available in the current repository documentation. If you require these technical details, please contact support.

Test the integration

Before relying on the webhook for production workflows, confirm that your endpoint is reachable.

1. Click the **Send Test Payload** button.
2. Check your server logs to confirm the test payload was received.
3. Review the **Recent deliveries** log in the DocuBot dashboard to see the response code returned by your server.

Manage delivery failures

If your endpoint is temporarily unavailable, DocuBot attempts to deliver the notification again.

1. Monitor the delivery status in your dashboard.
2. Notice that the system uses an **exponential backoff policy** for retries, increasing the time between attempts if your server continues to return errors.

3. Check the retry attempt count in the delivery logs to diagnose persistent connection issues.

Get help

If you encounter issues setting up your webhook integration or need specific payload schemas, please reach out to our support team:

- **Email**: support@ademero.com
- **Web**: https://www.ademero.com
- **Phone**: 863-937-0272

Code examples

You can use the following examples to understand how to configure your documentation sites, manage sync schedules, and integrate API references. Note that programmatic API endpoints for documentation management are not available in the public repository.

User authentication and authorization

DocuBot uses Google Sign-In for primary workspace access. To ingest private repositories, you must authorize GitHub access through the settings panel.

Example: GitHub OAuth connection status

```
{
  "githubConnected": true,
  "githubUsername": "octocat",
  "scope": "repo, read:user"
}
```

Expected result The dashboard displays your GitHub username and allows you to select private repositories as content sources.

Documentation site configuration

When you create a documentation project, you define its identity using a display name and a unique URL slug. The slug determines the public web address.

Example: Defining a project identity

```
{
  "displayName": "Cortexa Platform",
  "slug": "cortexa-platform"
}
```

Expected result Your documentation publishes to `docubot.cc/cortexa-platform` .

Sync schedule and frequency

You can automate documentation updates by setting a sync frequency. Updates occur at a specific local time based on your preferred time zone.

Example: Daily sync configuration

```
{
  "syncFrequency": "daily",
  "syncTimeLocal": "03:00",
  "syncTimezone": "America/New_York"
}
```

Expected result DocuBot pulls the latest code and regenerates your documentation every day at 3:00 AM EST.

API reference and OpenAPI integration

DocuBot automatically detects public OpenAPI or Swagger specifications within your connected repositories. These specifications are rendered into interactive documentation pages.

Example: OpenAPI specification detection

```
Input: openapi.json

Detection: Valid OpenAPI 3.0 specification found.
```

Expected result An "API Reference" entry appears in your documentation sidebar, linking to an interactive Scalar viewer at `/api-reference/openapi/openapi`.

Advanced documentation inputs

You can guide the AI generation process by providing specific scenarios or known issues. This ensures that your How-to guides and workaround articles address your most important user needs.

Example: Scenario focus input

```
- Authenticating with a bearer token

- Creating a new document via the dashboard

- Exporting data to CSV format
```

Expected result The generated "Core scenario how-to" guide explicitly includes detailed steps for these three workflows.

Contact and support

If you need assistance with configuration or have questions about the documentation engine, please use the following channels.

- **Support Portal**: https://www.ademero.com
- **Email**: support@ademero.com
- **Phone**: 863-937-0272

Next steps: Quick start guide or Conceptual overview

Conceptual overview

About DocuBot

DocuBot is a documentation engine that bridges the gap between technical source code and user-friendly help centers. It transforms your GitHub repositories into living, structured documentation sites that stay aligned with your latest code updates. Instead of manually writing and maintaining static pages, you use DocuBot to generate high-quality, searchable documentation tailored to specific audiences.

The DocuBot mental model

Most documentation tools focus on simple file-to-page conversion. DocuBot uses an audience-first philosophy. You start by identifying who you serve—such as end users, administrators, or developers—and the system maps those audiences to the specific document types they need.

By blending a universal voice with specialized document-type rules, DocuBot ensures that every page feels purposeful and cohesive. This approach reduces factual drift and ensures that your documentation reads as if a dedicated team wrote it, even as your codebase evolves.

Key concepts

To manage your documentation effectively, you should understand the three pillars of the DocuBot system:

- **Audiences**: These define who the documentation is for. You can select from End User, Power User, Administrator, and Developer. Your selection determines the default set of documents DocuBot generates.
- **Document types**: These are specific templates designed for different tasks. Examples include Quick Start Guides for onboarding, Troubleshooting Guides for resolving issues, and API References for technical integrations.
- **General styles**: These control the overall tone and voice of your documentation. You can choose styles like “Precision Technical” for exact, scan-friendly data or “Approachable Guided” for warm, step-by-step instructions.

How it fits into your workflow

DocuBot integrates directly into your existing development lifecycle to keep your help center current.

1. **Connect sources**: You connect one or more GitHub repositories to a single product documentation site. This allows you to aggregate information from multiple apps or services into one centralized location.
2. **Configure preferences**: You define your target audiences, refine the list of document types, and select a general style that matches your brand’s voice.
3. **Automated synchronization**: You set a sync schedule—daily, weekly, or monthly—at a specific local time. DocuBot automatically pulls the latest code, regenerates the content, and publishes the updates.
4. **Manual overrides**: You can provide custom inputs, such as specific scenario focuses or known issues, to prioritize certain workflows or workarounds in the generated guides.

Why static documentation matters

DocuBot produces static HTML output, which offers several advantages for your team and your users:

- **Performance and SEO**: Static pages load instantly and are highly visible to search engines, making it easier for users to find answers through external search.
- **Shareability**: Clean, SEO-friendly routes (URLs) allow your support team to share specific links with customers confidently.
- **Interactive API references**: If your repository contains OpenAPI or Swagger specifications, DocuBot renders them in a modern, interactive viewer for external developers.
- **Offline access**: You can export your entire documentation set as a professional PDF, complete with an automatic table of contents and bookmarks.

If you need assistance setting up your workspace or have questions about the documentation engine, contact support at support@ademero.com or visit https://www.ademero.com.

Key concepts

DocuBot turns your GitHub repositories into structured, static help centers. To manage your documentation effectively, you should understand these four foundational building blocks of the platform.

Documentation sites

A documentation site is the top-level container for your product's help center. It acts as the public home for your content, accessible through a unique URL slug (such as `docubot.cc/your-product`).

Unlike standard documentation tools that often link to a single repository, a DocuBot site can aggregate content from multiple sources. This allows you to present a unified experience to your users even if your product's code is spread across different repositories.

Example: You can create one "Finance Suite" documentation site that pulls information from your web dashboard repository, your mobile app repository, and your backend API repository simultaneously.

Audience-driven presets

DocuBot uses target audiences to automatically organize and prioritize your content. When you define who your readers are—such as End Users, Power Users, Administrators, or Developers—the system applies a preset that selects the most relevant document types for those groups.

While these presets provide a professional baseline, you can always refine the list by adding or removing specific document types to fit your project's unique needs.

Example: If you select the "Developer" audience, DocuBot automatically includes API References and Authentication Guides. If you select "End User," the system prioritizes Quick Start guides and core workflow tutorials.

Documentation styles

General styles act as a global overlay for your entire site. They control the voice, tone, and structural rules of every generated page. By selecting a style, you ensure that your documentation feels cohesive and professional, regardless of which team member wrote the code or which repository the data comes from.

Example: Choosing the "Precision Technical" style ensures your documentation uses exact terminology and structured tables for all settings. Switching to "Approachable Guided" transforms that same information into warmer, simpler language designed for non-technical readers.

Automated synchronization

Synchronization is the engine that keeps your documentation aligned with your latest code. Instead of manually updating help articles every time you ship a feature, you can set an automated schedule. DocuBot connects to your GitHub repositories and refreshes the content based on your preferred frequency—daily, weekly, or monthly.

Example: You can schedule a daily sync for 3:00 AM in your local time zone. Every morning, your documentation site automatically reflects the latest changes pushed to your GitHub branches the previous day.

Support and assistance

If you have questions about these concepts or need help configuring your workspace, you can reach out through the following channels:

- **Email:** support@ademero.com
- **Help Portal:** https://www.ademero.com
- **Phone:** 863-937-0272

Architecture overview

DocuBot operates as a coordinated system that transforms your source code into structured, audience-driven documentation. The architecture ensures that your help centers remain synchronized with your codebase while providing high-performance, search-optimized experiences for your readers.

Core system components

The DocuBot ecosystem consists of four primary functional areas that work together to manage and deliver your documentation.

- **Centralized dashboard**: This is your management workspace where you define your products, configure repository sources, and monitor the status of your documentation sites.
- **GitHub integration layer**: This component handles the secure ingestion of content from your repositories. It manages the connection to both public and private sources to ensure the documentation engine has the necessary context.
- **Documentation engine**: The core processing unit that analyzes your code. It applies your selected audiences, document types, and general styles to generate purposeful content.
- **Published documentation sites**: The final output of the system. These are high-performance, static sites that include interactive API viewers and searchable content.

The documentation lifecycle

Your documentation moves through a specific lifecycle from the moment you connect a repository to the final publication.

1. **Synchronization**: DocuBot checks your repositories based on your defined schedule (daily, weekly, or monthly). You can also trigger manual updates through the dashboard.
2. **Transformation**: The engine analyzes your source code and transforms technical implementation details into structured document types, such as Quick Start Guides or Troubleshooting articles.
3. **Style application**: The system layers your chosen general style with quadrant-specific guidelines to ensure the voice and tone remain consistent across the entire doc set.
4. **Publication**: DocuBot generates static web pages and PDF exports, then deploys them to your unique URL slug.

Data flow and integration boundaries

DocuBot maintains a clear separation between your management environment and the public-facing documentation to ensure security and performance.

- **Secure authorization**: For private repositories, DocuBot uses GitHub OAuth to gain read-only access to your code without requiring you to manage individual access keys.
- **Content ingestion**: The system pulls the latest state of your default branch during every sync cycle. It specifically looks for code patterns, README files, and API specifications.
- **Workspace separation**: Your internal configuration—such as sync times, scenario priorities, and known issues—lives within your private workspace. Only the generated documentation is exposed to the public.
- **Update triggers**: You control when data flows from your repo to your docs through automated windows or manual overrides.

API reference architecture

Technical specifications receive specialized handling within the DocuBot ecosystem to provide a modern developer experience.

- **Automatic detection**: DocuBot automatically scans your connected sources for OpenAPI and Swagger specification files.
- **Interactive rendering**: When the engine finds a specification, it integrates an interactive reference tool into your documentation site. This allows developers to explore endpoints and test requests directly in the browser.
- **Multi-source aggregation**: If your product spans multiple repositories with different APIs, DocuBot aggregates them under a single documentation slug, providing a unified entry point for your integrators.

Design rationale

The system is built as a static documentation engine rather than a runtime generator to prioritize the needs of your end users and support teams.

- **Search engine optimization**: Static HTML ensures that every page is easily indexable by search engines, making your help content discoverable for users seeking answers.
- **Performance and reliability**: Pre-generated pages load instantly and remain available even if your source repositories are undergoing maintenance.
- **Version alignment**: By generating docs during a sync cycle, DocuBot ensures that the published content reflects a specific, stable version of your code.
- **Link integrity**: Static routes provide reliable, permanent URLs that your customer support team can share with confidence.

If you have questions about how DocuBot handles your specific repository structure, please contact our team at support@ademero.com, visit https://www.ademero.com, or call 863-937-0272.

Glossary

This glossary defines the essential terms and concepts you encounter while using DocuBot. Use this reference to understand how the platform organizes, generates, and delivers your documentation.

Core platform terms

Documentation site The public-facing help center generated by DocuBot. It hosts your guides, references, and tutorials at a dedicated web address.

Repository source A GitHub repository you connect to a documentation site. DocuBot analyzes the code and metadata within these sources to generate content.

URL slug The unique identifier used in your documentation site's web address. For example, in `docubot.cc/my-product` , "my-product" is the slug.

Workspace Your private dashboard where you manage your documentation sites, repository connections, and account preferences.

Configuration and customization

Audience The target reader group for your documentation. You can select from End User, Power User, Administrator, or Developer. Your selection determines the default set of document types DocuBot generates.

Document type A specific template or guide format designed for a particular purpose, such as a Quick Start Guide, Troubleshooting Guide, or API Reference.

General style The overall tone, voice, and structural rules applied to your entire documentation set. Options include "Precision Technical" for exactness or "Approachable Guided" for a warmer, simpler tone.

Known issues Custom inputs you provide regarding documented bugs or limitations. DocuBot uses these to generate specific workaround and knowledge base articles.

Scenario focus Specific user workflows or tasks you prioritize. Providing these inputs helps the AI tailor how-to guides to your most important use cases.

Automation and delivery

OpenAPI reference An interactive, searchable API viewer rendered automatically from Swagger or OpenAPI specifications found in your repository.

Role detection The automated process where DocuBot identifies the purpose of a repository source, such as a "Primary UI" for your main application or "API Reference" for your backend services.

Static output Documentation delivered as pre-rendered HTML files. This ensures your help center is fast, reliable, and optimized for search engines (SEO).

Sync schedule The automated frequency at which DocuBot refreshes your documentation. You can configure this to run daily, weekly, or monthly at a specific time.

Support and integration

GitHub OAuth The secure authorization protocol you use to grant DocuBot access to your GitHub account. This is required to generate documentation from private repositories.

PDF export A downloadable version of your documentation site. It includes an automated table of contents and bookmarks for easy offline navigation.

Support contacts The specific email addresses, portal URLs, and phone numbers you configure in your settings. These details are automatically embedded into your "Contact and support" pages. If you need assistance with DocuBot itself, you can reach out to support@ademero.com or visit https://www.ademero.com.

Best practices guide

High-quality documentation depends on how you configure your DocuBot workspace. By aligning your repository settings, audience selections, and style choices with your actual product structure, you ensure that the generated help center is accurate, findable, and useful for your readers.

Optimizing repository roles

When your product spans multiple GitHub repositories, assigning the correct role to each source is the most important step for high-quality generation. These roles provide the necessary context for the AI to understand how different parts of your codebase relate to each other.

- **Primary UI**: Assign this role to your main application. DocuBot uses this repository as the anchor for your documentation, prioritizing its features and workflows in the generated guides.
- **UI App**: Use this for secondary experiences, such as an admin console, a mobile app, or a desktop client that complements the primary application.
- **API Reference**: Assign this to repositories that contain your OpenAPI or Swagger specifications. This ensures your interactive API documentation is correctly rendered and linked.
- **Reference**: Use this for supporting libraries or non-UI repositories that provide essential background information but aren’t the main focus of user workflows.

Effective audience and doc type selection

DocuBot uses your selected audiences to auto-select a preset of document types. While these presets are a great starting point, you should refine them to match your specific user base.

- **Avoid information overload**: Only select audiences that actually exist for your product. For example, if you don’t offer a public API, deselecting the “Developer” audience keeps your documentation site focused and prevents the generation of empty reference sections.
- **Customize the list**: You can add or remove specific document types after selecting your audiences. If a specific guide like “Keyboard shortcuts” isn’t relevant to your app, remove it to keep your sidebar clean.
- **Mind the reset**: Be aware that changing your audience selection after you have customized your document types will reset the list to the default preset for those audiences.

Leveraging advanced doc inputs

The “Advanced doc inputs” section in your repository settings allows you to provide specific context that the AI cannot always infer from code alone.

- **Scenario Focus**: Use this area to list the specific user workflows you care about most. Listing scenarios like “How to invite a team member” or “How to export a report” ensures these tasks are prioritized in your How-to guides.
- **Known Issues**: Documenting known limitations or temporary bugs here allows DocuBot to generate helpful workaround articles and KB entries, reducing the burden on your support team.
- **Support Contacts**: Ensure your workspace profile includes your support email, portal URL, and phone number. DocuBot automatically injects these details into the “Contact and support” page so users know exactly how to get help.

Maintaining documentation freshness

Your documentation is only as good as its last update. Configure your sync schedule to balance accuracy with your team’s development velocity.

- **Align with deployments**: Set your sync time to occur shortly after your team typically finishes their daily or weekly deployments. This ensures your docs reflect the latest code changes.
- **Frequency selection**: Use daily syncs for rapidly evolving products. For stable projects, a weekly or monthly sync may be sufficient.
- **Manual updates**: If you make a major change and need the docs updated immediately, use the “Rebuild documentation” button on your repository card. Note that manual syncs are subject to daily limits.

Style selection for your brand

The general style you choose controls the voice and tone of every page in your documentation site.

- **Precision Technical**: Best for developer-focused products or complex administrative tools where exact values, constraints, and structured tables are more important than narrative.
- **Approachable Guided**: Ideal for non-technical end users. This style uses warm, supportive language and focuses on quick wins and simple, single-action steps.
- **DocuBot Default**: A balanced choice for products with a mixed audience. it provides clear, direct, and professional guidance that works well for both beginners and power users.

If you need assistance optimizing your documentation setup, contact our support team at support@ademero.com, visit https://www.ademero.com, or call 863-937-0272.

Security and compliance overview

DocuBot protects your source code and documentation through a security-first architecture. By focusing on ephemeral processing and static output, the system minimizes the risk to your intellectual property while providing high-quality help centers.

Security architecture overview

DocuBot uses a high-level security model designed to protect your data through every stage of the documentation lifecycle. The system follows a philosophy of ephemeral processing, meaning it only accesses your repository content during the active generation window.

The primary advantage of this model is the static HTML output. Because your documentation site is pre-generated, the public-facing help center doesn't require a live connection to your source code or internal databases. This separation creates a natural security barrier between your development environment and your end users.

Data protection and privacy

We handle your repository content with strict privacy controls. When you trigger a sync, DocuBot retrieves the necessary code segments to analyze your product's functionality. Once the documentation is generated, these temporary segments are cleared from the processing environment.

Your data is protected using the following methods:

- **Encryption in transit**: All communication between your browser, GitHub, and DocuBot is encrypted using standard TLS/SSL protocols.
- **Encryption at rest**: Generated documentation artifacts and workspace settings are stored using industry-standard encryption provided by our cloud infrastructure.
- **Secure hosting**: Your static documentation sites are hosted in isolated environments to prevent cross-site vulnerabilities.

Information regarding specific data retention policies for generated artifacts is not available in the current repository.

Authentication and access control

You manage your DocuBot workspace through secure authentication providers. The system uses Google Sign-In to verify your identity, ensuring that only authorized users can access the dashboard or modify repository settings.

To document private repositories, you must authorize DocuBot via GitHub OAuth. We follow the principle of least privilege, requesting only the scopes necessary to read your repository content. You can revoke this access at any time through your GitHub account settings.

Within the dashboard, you can manage permissions for your documentation sites, including setting up URL slugs and choosing which audiences can view specific document types.

Compliance and infrastructure standards

DocuBot is built on modern cloud infrastructure that adheres to global security best practices. We perform regular security updates and monitoring to protect against emerging threats.

While the platform follows standard encryption and security protocols, information regarding specific SOC2 or ISO certifications is not available at this time. The system relies on the underlying security controls of our cloud providers to maintain high availability and network-level protection.

Security support and reporting

We take security inquiries and vulnerability reports seriously. If you have questions about our security posture or need to report a potential issue, please use our official support channels.

You can reach the security team through the following methods:

- **Email**: support@ademero.com
- **Support Portal**: https://www.ademero.com
- **Phone**: 863-937-0272

When reporting a vulnerability, please include a detailed description of the issue and steps to reproduce it so our team can investigate promptly.

How it works

DocuBot is more than a simple text generator; it's a documentation engine designed to understand the relationships between your codebases and your users. By analyzing your source code and applying a layered style system, DocuBot creates structured, static help centers that read as if they were written by a human technical writer.

Multi-repository aggregation and roles

DocuBot treats your documentation site as a unified product, even if that product spans multiple GitHub repositories. When you connect your sources, the engine assigns a functional role to each repository to guide the AI's focus:

- **Primary UI**: This is the main application experience. DocuBot prioritizes this code to understand the core user journey.
- **UI App**: These are secondary experiences, such as an admin console or a mobile app, that complement the primary product.
- **API Reference**: Repositories containing OpenAPI or Swagger specifications are treated as technical data sources for developer documentation.
- **Reference**: Supporting codebases that provide context but aren't the primary focus for end-user guides.

You can override these roles at any time to ensure the AI understands exactly which parts of your codebase deserve the most attention.

The documentation generation pipeline

The lifecycle of a documentation sync follows a rigorous process to ensure accuracy and prevent the "hallucinations" common in standard AI tools.

1. **Repository scanning**: DocuBot identifies UI signals, functional modules, and API specifications within your connected repositories.
2. **Structural planning**: Before writing any content, the engine creates a detailed plan for every document type you selected. This plan identifies the intent of each section and the specific evidence in the code that supports it.
3. **Content writing**: Using the structural plan as a map, the engine writes the documentation using your selected voice and tone.
4. **Static compilation**: The final content is compiled into high-performance HTML pages, ensuring your documentation is ready for public consumption immediately.

Layered style fusion engine

To keep your documentation cohesive, DocuBot uses a "style fusion" engine that combines three distinct layers of guidance:

- **General Styles**: This is your overall brand voice. Whether you choose "Precision Technical" for developers or "Approachable Guided" for new users, this layer controls the vocabulary and attitude of the writing.
- **Quadrant Guidelines**: Based on the Diátaxis framework, this layer determines the structure of the page. It ensures that a tutorial feels like a lesson, while a reference page stays focused on facts.
- **Document Type Templates**: These are specific rules for pages like "Quick Start Guides" or "FAQs." They ensure that every page of a certain type contains the information users expect to find.

AI context management and accuracy

DocuBot maintains technical accuracy through a system called context budgeting. Because codebases can be massive, the engine calculates a "budget" for each repository based on its assigned role.

The primary application receives the largest share of the budget, allowing the AI to see the most detail. Shared context packs then allow the documentation for one repository to understand dependencies in another. This means your "Integration Guide" can accurately describe how your frontend interacts with your backend API without losing track of the details in either.

Design rationale: Static vs. runtime delivery

DocuBot generates static documentation rather than rendering content on demand. This approach provides several key benefits for your team and your users:

- **Performance**: Pre-generated HTML pages load nearly instantly, providing a better experience for users seeking quick answers.

- **SEO optimization**: Because the pages are static, search engines can easily crawl and index every guide, making your help center easier to find via Google.
- **Reliable linking**: Every page has a real, permanent URL. Your support team can share links to specific sections with confidence, knowing the content is always there.

Interactive API rendering

When DocuBot detects OpenAPI or Swagger specifications, it handles them differently than standard prose. Instead of just writing about your API, it integrates an interactive reference viewer. This allows developers to explore endpoints, view data models, and even test live API calls directly from your documentation site. These technical references are automatically placed in a dedicated section of your sidebar for easy access.

If you encounter issues with your documentation sync or need help refining your style settings, contact our team at support@ademero.com or visit https://www.ademero.com.

FAQ

Find quick answers to common questions about setting up your workspace, managing repository synchronization, and customizing your documentation output.

General questions

What is DocuBot

DocuBot is an automated platform that transforms your GitHub repositories into structured, high-quality help centers. It uses AI to analyze your source code and generate static documentation that is optimized for search engines and technical accuracy.

How does the AI generation process work

The system analyzes your codebase to understand your product's features, architecture, and workflows. It then applies a layered style model—combining your selected general style with specific document-type rules—to write content tailored for your chosen audiences.

Is the documentation static or dynamic

All documentation is generated as static HTML. This ensures high performance, reliability, and excellent SEO visibility for your help center.

Setup and repository management

Can I group multiple repositories together

Yes. DocuBot allows you to aggregate multiple GitHub repositories under a single product documentation site. This is ideal for products that span across different apps, APIs, and reference libraries.

How do I connect a private repository

To connect a private repository, you must authorize DocuBot via GitHub OAuth. This provides the system with the necessary permissions to read your code and generate documentation without exposing your source files to the public.

What is a URL slug

A URL slug is the unique part of the web address that identifies your documentation site (e.g., `docubot.cc/your-product-name`). You define this slug during the initial setup of your documentation site.

Syncing and updates

How often does my documentation update

You can configure an automated synchronization schedule based on your needs. DocuBot supports daily, weekly, and monthly sync frequencies at a specific local time of your choosing.

Can I trigger a manual update

Manual syncs are available for paid accounts. This allows you to rebuild and publish your documentation immediately after pushing significant changes to your repository.

What happens if there are no code changes

If the system detects that the latest commit matches the last synced version, it will skip the generation process to save resources while still recording the successful check in your dashboard.

Customizing documentation styles

What are General Styles

General Styles control the overall tone and structural rules of your documentation. You can choose from options like **Precision Technical** for developer-focused content or **Approachable Guided** for a warmer, onboarding-friendly experience.

How do audiences impact my documentation

When you select target audiences (such as End User, Administrator, or Developer), DocuBot automatically pre-selects a relevant set of document types. For example, selecting "Developer" will prioritize API references and quick-start guides.

What is Scenario Focus

Scenario Focus is an advanced input field where you can list specific user workflows you want the AI to prioritize. This ensures your "How-to" guides cover the tasks most important to your users.

API and developer reference

How are API references rendered

DocuBot automatically detects OpenAPI and Swagger specifications within your repositories. It renders these into a dedicated, interactive API viewer powered by Scalar, allowing developers to explore endpoints and payloads directly.

Does DocuBot support interactive documentation

Yes. The API reference section is fully interactive, providing a modern interface for developers to test and understand your public interfaces.

Support and account management

What are the limits for free accounts

Free accounts are limited to tracking one repository and are restricted to a monthly synchronization schedule. Manual syncs and more frequent updates require a paid subscription.

How do I get help

If you encounter issues or have questions not covered here, you can reach out to our support team through the following channels:

- **Email**: support@ademero.com
- **Web**: https://www.ademero.com
- **Phone**: 863-937-0272

How do I delete my account

You can manage your account and data through the Profile section in your dashboard. Deleting your account will remove all tracked repositories, generated documentation, and stored integration tokens.

Design rationale

DocuBot is designed to solve the disconnect between rapidly evolving source code and the documentation that supports it. By prioritizing static output, audience-specific architecture, and AI-driven synchronization, the system ensures that your help centers remain accurate, performant, and relevant to the people who use your products.

Static documentation strategy

DocuBot prioritizes pre-generated, static HTML over runtime or on-demand rendering. This decision is driven by three primary factors: performance, reliability, and search engine optimization (SEO).

Static pages load instantly because they require no server-side processing or database lookups at the moment a user requests them. This ensures a high-quality experience for your customers, even under heavy traffic. Furthermore, static output provides real, linkable routes that search engines can easily crawl and index. By avoiding runtime generation, we eliminate the risk of “documentation downtime” or slow response times that often plague dynamic help systems.

Audience-centric information architecture

Most documentation tools follow a flat file structure or mirror the organization of the source code. DocuBot intentionally departs from this by structuring content around specific user roles and document types.

We use audience-specific presets—such as End User, Administrator, and Developer—to determine which types of information are most valuable. This ensures that a developer looking for an API reference isn’t forced to sift through basic onboarding tutorials. Our layered style model allows us to apply a consistent general voice across your entire site while still respecting the unique structural requirements of different content quadrants, such as how-to guides or conceptual explanations.

Multi-repository aggregation

Modern software products often span multiple repositories, including separate codebases for web apps, mobile clients, and backend APIs. DocuBot allows you to aggregate these disparate sources into a single, unified product documentation site.

This approach reflects the reality of the user experience. Your customers view your product as a single entity, regardless of how many repositories exist behind the scenes. By centralizing documentation, DocuBot enables cross-referencing between different components of your ecosystem and provides a cohesive “source of truth” for your entire product line.

AI-driven content alignment

The core of DocuBot’s value lies in its ability to generate documentation directly from your source code using artificial intelligence. This ensures that your documentation stays synchronized with the latest features and changes your team ships.

Manual documentation is prone to factual drift and human error. By using AI to analyze the code itself, DocuBot maintains technical accuracy and reduces the manual toil required to keep help centers current. The system is designed to learn from your code, translating complex logic into structured, human-readable guides that follow professional documentation standards.

Interactive API reference selection

For developer-focused documentation, we have selected Scalar as the primary engine for rendering OpenAPI specifications. Scalar provides a modern, dedicated API viewer that is both interactive and highly readable.

We chose this integration because it offers a superior developer experience compared to traditional, static API lists. It allows integrators to explore endpoints, view request/response models, and interact with your API directly within the documentation suite. This reduces the friction for external developers and improves the overall adoption of your public interfaces.

Contact and support

If you have questions about the architectural decisions or the behavior of the documentation engine, please reach out to our team for assistance.

- **Email**: support@ademero.com
- **Support Portal**: https://www.ademero.com

- **Phone**: 863-937-0272

API reference

The DocuBot API allows you to programmatically manage your documentation sites, configure repository sources, and trigger synchronization jobs. All API requests require authentication and return data in JSON format.

Authentication

You must authenticate every request to the DocuBot API using a Bearer token. This token is associated with your Google Sign-In credentials.

Include the token in the `Authorization` header of your HTTP requests:

```
Authorization: Bearer <YOUR_TOKEN>
```

Create repository

Register a new product documentation site and connect its source repositories.

- **Endpoint:** `POST /api/repos`
- **Format:** `application/json`

Request body

Field	Type	Description
`displayName`	string	The internal name for the repository workspace.
`slug`	string	The unique URL path for your documentation (e.g., `my-product`).
`sources`	array	A list of objects containing `repoUrl` (GitHub URL) and optional `label` .
`syncFrequency`	string	How often to update: `daily` , `weekly` , or `monthly` .
`generalStyleKey`	string	The visual and tonal style, such as `docubot-default` .
`audiences`	array	Target groups like `End User` , `Admin` , or `Developer` .
`docTypes`	array	Specific document keys to generate.

Update repository

Modify the configuration, audiences, or focus areas for an existing documentation site.

- **Endpoint:** `PATCH /api/repos/{repoId}`
- **Format:** `application/json`

Request body

Field	Type	Description
`displayName`	string	Updated name for the workspace.
`audiences`	array	Updated list of target audiences.
`docTypes`	array	Updated list of document types to include.
`scenarioFocus`	array	A list of specific user workflows to prioritize in guides.
`knownIssues`	array	A list of issues to include in workaround articles.

Trigger manual sync

Request an immediate documentation rebuild outside of your automated schedule.

- **Endpoint:** `POST /api/repos/{repoId}/sync`
- **Format:** `application/json`

Request body

Field	Type	Description
`rebuildOnly`	boolean	If `true` , regenerates site files without pulling new code changes.

Note: Manual syncs are subject to daily limits based on your account tier.

GitHub connection

Initiate the OAuth flow to grant DocuBot access to your private repositories.

- **Endpoint:** `POST /api/github/connect`

Response

The API returns an `authorizeUrl` . You must redirect your browser to this URL to complete the GitHub authorization process.

Invite request

Request access to the DocuBot platform if your account is not yet authorized.

- **Endpoint:** `POST /api/invite-request`

This endpoint records your request for review by the DocuBot administration team.

Account management

Permanently remove your account and all associated documentation data.

- **Endpoint:** `POST /api/account/delete`
- **Format:** `application/json`

Request body

Field	Type	Description
`confirm`	string	You must provide the exact phrase: `delete my account` .

Warning: This action is irreversible and deletes all tracked repositories and generated documentation.

Support

If you encounter issues with the API or need assistance with integration, contact our support team:

- **Email:** support@ademero.com
- **Web:** https://www.ademero.com
- **Phone:** 863-937-0272

Settings reference

This reference provides a comprehensive list of configurable options within the DocuBot dashboard. Use these settings to manage your documentation site identity, content strategy, and automated synchronization behavior.

Workspace identity

These settings define how your project is organized internally and how it appears to the public.

Setting	Description	Impact
Display name	The internal name for the repository workspace.	Used for organization within your DocuBot dashboard.
URL slug	A unique identifier that forms the public web address.	Determines the live URL path (e.g., `docubot.cc/your-slug`).
Live URL preview	A read-only preview of the final documentation path.	Allows you to verify the public address before publishing.

Content strategy

Configure who your documentation serves and how the AI engine structures the information.

Setting	Description	Impact
Audiences	Select target readers: End User, Power User, Admin, or Developer.	Auto-selects a recommended preset of document types.
Document types	Specific templates like Quick Start Guides or API References.	Determines which pages are generated and published.
General style	Selectable voices such as “Precision Technical” or “Approachable Guided.”	Controls the overall tone, reading level, and structural rules.

Automation and scheduling

Manage when and how often DocuBot pulls the latest changes from your connected repositories.

Setting	Description	Impact
Sync frequency	Choose between Daily, Weekly, or Monthly updates.	Determines the interval for automated documentation refreshes.
Update time	The specific time of day for the sync to execute.	Ensures updates happen during low-traffic or preferred windows.
Time zone	Your preferred regional time alignment.	Aligns the update time and scheduling to your local clock.

AI generation inputs

Provide specific context to the AI engine to improve the relevance of generated guides and articles.

Setting	Description	Impact
Scenario priorities	A list of specific user workflows or tasks.	Prioritizes these scenarios in How-to and Workflow guides.
Known issues	A list of documented bugs or system limitations.	Used to generate Workaround and KB articles.

Support and metadata

These details populate the public-facing contact and support pages for your documentation site.

Setting	Description	Impact
Support email	The primary email address for user help.	Appears on the “Contact and support” page.
Support portal URL	The link to your external help desk or ticket system.	Provides a direct link for users to escalate issues.
Support phone	The contact number for live support.	Listed as an available support channel for your users.

Contact information

If you need assistance with these settings or your DocuBot workspace, contact our support team:

- **Email**: support@ademero.com
- **Web**: https://www.ademero.com
- **Phone**: 863-937-0272

Error code reference

This reference guide helps you identify and resolve common errors encountered while configuring your workspace, connecting repositories, or synchronizing documentation.

Authentication and access errors

These errors typically occur when connecting your Google account or authorizing DocuBot to access your GitHub repositories.

Error message	Cause	Resolution
Invalid OAuth state	The authentication session timed out or the security token expired.	Restart the connection process from your profile or settings page. Ensure you complete the GitHub authorization within 15 minutes.
GitHub authorization required	You're attempting to add or sync a private repository without an active GitHub connection.	Go to **Settings** and click **Connect GitHub** to authorize access to your private repositories.
Unable to verify user	The system couldn't confirm your account details during the sign-in process.	Sign out and sign back in using your Google account. Ensure your email is verified.

Repository configuration errors

Use this section to resolve issues related to setting up a new documentation site or modifying existing repository sources.

Error message	Cause	Resolution
That slug is already in use	The URL slug you chose is already taken by another documentation site.	Choose a unique slug using lowercase letters, numbers, and dashes.
Invalid repo URL	The provided URL isn't a valid GitHub repository link.	Verify the URL in your browser. It must follow the format `https://github.com/owner/repo`.
Free accounts can only track one repository	You've reached the project limit for the free tier.	Upgrade to a paid plan to add more repositories, or delete your existing repository before adding a new one.
Unknown repo role selected	The manually assigned role for a repository source is invalid.	Select one of the standard roles: Primary UI, UI app, API reference, or Reference only.

Sync and generation errors

These errors occur during the automated or manual process of turning your code into documentation.

Error message	Cause	Resolution
Manual sync limit reached for today	You've exceeded the daily limit for manual documentation rebuilds.	Wait for your next scheduled daily sync or try again tomorrow.
A sync is already running for this repo	A documentation build is currently in progress for this project.	Wait for the active sync to finish. You can monitor the progress on the repository card in your dashboard.
Repository context is too large	The repository contains too much data for the AI engine to process in one pass.	Reduce the number of files by ignoring non-essential directories or contact support for high-capacity processing.
Doc generation failed	An internal error occurred during the AI writing or site building phase.	Check that your repository is still accessible and contains valid source code, then try a manual rebuild.

Interactive API reference errors

Troubleshoot issues related to the ingestion of OpenAPI specifications and the rendering of interactive API viewers.

Error message	Cause	Resolution
OpenAPI file not detected	The system couldn't find a valid OpenAPI or Swagger specification in the repository.	Ensure your specification file is in JSON or YAML format and includes the `openapi:` or `swagger:` version tag.
Scalar API viewer failed to load	The interactive documentation component couldn't retrieve the specification file.	Verify that the specification file is valid and accessible within your repository. Check your browser console for connectivity issues.

Contacting support

If you encounter an error not listed here or cannot resolve an issue using the steps provided, please reach out to our support team for assistance.

- **Email**: support@ademero.com
- **Support Portal**: https://www.ademero.com
- **Phone**: 863-937-0272

When contacting support, please include the repository name, the specific error message you received, and the time the error occurred.

Keyboard shortcuts

Use keyboard shortcuts to navigate your DocuBot workspace and documentation sites more efficiently. These shortcuts help you jump to search bars and move through interactive API references without using your mouse.

Global shortcuts

The following shortcut is available across the DocuBot dashboard and all generated documentation sites.

Action	macOS	Windows / Linux
Activate search	`Cmd + K`	`Ctrl + K`

Documentation viewer

When viewing your published documentation, you can use these additional shortcuts to navigate pages and interactive API references.

Action	Shortcut
Open search	`/` or `K`
Close search or modals	`Esc`
Navigate search results	`Up` / `Down` arrows
Select search result	`Enter`

API reference navigation

Interactive API references powered by Scalar support standard keyboard navigation for exploring endpoints and schemas.

- **Tab**: Move between navigation links, request parameters, and code snippets.
- **Enter / Space**: Expand or collapse endpoint details and folders.
- **Esc**: Close full-screen views or interactive overlays.

Platform notes

DocuBot adjusts modifier keys based on your operating system to ensure a consistent experience.

- **macOS**: Use the **Command** key (`Cmd`) for global shortcuts.
- **Windows / Linux**: Use the **Control** key (`Ctrl`) for global shortcuts.

If you encounter issues with shortcuts or have suggestions for new navigation patterns, please contact our support team at support@ademero.com or visit https://www.ademero.com.

Next steps

- Manage your repositories
- Update workspace settings

Feature matrix

DocuBot offers different tiers to match the scale of your projects. Whether you are documenting a single open-source tool or a complex product suite spanning multiple repositories, you can select the plan that fits your workflow.

Plan comparison

Use the following table to compare the core limits and capabilities of the Free and Paid tiers.

Feature	Free Plan	Paid Plan
Tracked repositories	1	Multiple
Automated sync frequency	Monthly	Daily, Weekly, or Monthly
Manual sync requests	Not available	Included
Multi-repo aggregation	Not available	Included
Static HTML output	Included	Included
PDF documentation exports	Included	Included

Documentation generation features

Every DocuBot plan includes access to our core AI-driven documentation engine. Regardless of your tier, you receive high-quality, structured help centers generated directly from your source code.

- **AI-driven content:** Automatically generates tutorials, how-to guides, and reference material.
- **Static HTML output:** Produces high-performance pages optimized for search engine visibility.
- **Audience presets:** Choose from End User, Power User, Administrator, or Developer presets to auto-select relevant document types.
- **Selectable styles:** Apply different voices to your documentation, such as “Precision Technical” or “Approachable Guided.”
- **Interactive API reference:** Automatically detects and renders OpenAPI/Swagger specifications using a modern, interactive viewer.
- **PDF exports:** Generates downloadable versions of your documentation with automatic tables of contents and bookmarks.

Sync and automation limits

DocuBot keeps your documentation aligned with your code through automated synchronization. The frequency of these updates depends on your plan level.

- **Free plan sync:** Your documentation site refreshes automatically once per month.
- **Paid plan sync:** You can configure an automated update schedule that runs daily or weekly at a specific local time.
- **Manual updates:** Paid users can trigger an immediate rebuild of their documentation through the dashboard. This is useful when you ship a major update and want the docs to reflect the changes instantly.

Account and integration support

Managing your workspace and connecting your code sources is straightforward on all plans.

- **Authentication:** Sign in securely to the DocuBot dashboard using your Google account.
- **Public repositories:** Connect any public GitHub repository without additional authorization.
- **Private repositories:** Document private code by connecting your GitHub account via OAuth. This allows DocuBot to securely access your private source files during the generation process.

Contact support

If you have questions about plan limits or need help choosing the right tier for your team, reach out to us through the following channels:

- **Email:** support@ademero.com
- **Phone:** 863-937-0272
- **Web:** https://www.ademero.com

Permissions and roles reference

This reference describes the access levels and permissions available within DocuBot. Use this guide to understand how roles govern your ability to manage documentation sites, connect repositories, and configure synchronization settings.

User roles

DocuBot uses three primary roles to manage access to the dashboard and the generated documentation.

Role	Description
Owner	The individual who creates the documentation site. Owners have full administrative control over the site configuration, repository sources, and billing settings.
Authorized User	A user who has been granted access to the DocuBot dashboard. These users can manage repositories and trigger updates based on the organization’s allowlist requirements.
Public Viewer	Any external user who accesses the published documentation URL. Viewers can read all generated content and download PDF exports but cannot access the dashboard or configuration settings.

Permissions matrix

The following table outlines the specific actions available to each role within the platform.

Action	Owner	Authorized User	Public Viewer
Create documentation sites	Yes	Yes	No
Delete documentation sites	Yes	No	No
Connect GitHub repositories via OAuth	Yes	Yes	No
Configure sync schedules and frequencies	Yes	Yes	No
Trigger manual documentation rebuilds	Yes	Yes	No
Modify audience and document type selections	Yes	Yes	No
View generated documentation	Yes	Yes	Yes
Download PDF exports	Yes	Yes	Yes

Access and authorization

DocuBot relies on secure, industry-standard authentication methods to verify your identity and protect your repository data.

Account authentication

You must sign in to the DocuBot dashboard using **Google Sign-In**. Access to the workspace is governed by a domain-based allowlist. If your email domain is not on the authorized list, you will not be able to access the dashboard even after a successful Google sign-in.

Repository access

To generate documentation from private repositories, you must authorize DocuBot via **GitHub OAuth**. This connection allows DocuBot to securely retrieve your code for the purpose of documentation generation. Public repositories do not require this authorization.

Limitations

Granular team-based permissions or sub-administrator roles are not currently available. All authorized users within a workspace share the same management capabilities for the repositories they track.

Contact support

If you need to adjust your domain allowlist or require assistance with role assignments, please contact our support team:

- **Email**: support@ademero.com
- **Web**: https://www.ademero.com
- **Phone**: 863-937-0272

Limits and quotas

This reference outlines the operational limits and account quotas for the DocuBot platform. These constraints ensure system stability and provide a consistent experience across all documentation sites.

Account and repository limits

Your account type determines how many projects and sources you can manage within the DocuBot dashboard.

Feature	Free Tier	Paid Tier
Tracked repositories	1 repository	Multiple repositories
Documentation sites	1 site	Multiple sites
Repository sources	Single source per site	Multiple sources per site

Free accounts are restricted to a single repository source. If you need to aggregate multiple repositories under one documentation site or manage several distinct products, you must upgrade your account.

Synchronization and update quotas

DocuBot uses synchronization jobs to pull the latest code from GitHub and regenerate your documentation. The frequency and availability of these jobs depend on your subscription.

Automated sync frequency

- **Free Tier:** Automated updates occur once per month.
- **Paid Tier:** You can configure automated updates to run on a daily, weekly, or monthly schedule at a specific local time.

Manual sync limits

Manual synchronization allows you to trigger an immediate update outside of your scheduled window.

- **Availability:** Manual sync is a premium feature and is not available on the free tier.
- **Daily Limit:** You can request up to 5 manual synchronizations per repository every 24 hours.

Content and integration constraints

Technical limits apply to the amount of data DocuBot processes to maintain high-quality AI generation and fast build times.

- **API Specifications:** A single documentation site can ingest and render up to 50 individual OpenAPI or Swagger specifications.
- **Processing Budgets:** Repository context is subject to a token budget for AI analysis. For products with extremely large codebases, DocuBot prioritizes content based on the assigned repository role (such as Primary UI or API Reference) to ensure the most critical information is documented.

Managing limits

If you reach a quota or require higher limits for your organization, use the following methods to resolve the issue.

Upgrading your account

You can increase your repository count and unlock daily synchronization by upgrading your plan through the account settings in your dashboard.

Quota resets

Manual sync limits reset daily. If you reach the limit for a specific repository, you must wait for the daily reset or wait for your next scheduled automated synchronization to occur.

Contacting support

If you encounter unexpected errors related to system limits or require custom quotas for large-scale deployments, contact the support team:

- **Email:** support@ademero.com
- **Web:** https://www.ademero.com
- **Phone:** 863-937-0272

Release notes

Stay informed about the latest updates, new features, and improvements to the DocuBot platform. We update this page regularly as we release new capabilities and fixes to help you manage your product documentation more effectively.

Version 1.0.0

This is the initial public release of the DocuBot documentation engine. This version establishes the core foundation for turning your code repositories into structured, high-quality help centers.

New features

- **Interactive API references**: We integrated Scalar to provide modern, interactive API documentation. If your repository contains OpenAPI or Swagger specifications, DocuBot automatically renders them into a dedicated viewer.
- **Automated PDF exports**: You can now download a complete PDF version of your documentation. These exports include an automatically generated table of contents and bookmarks for easy navigation.
- **GitHub OAuth support**: You can securely connect your GitHub account to authorize DocuBot to access and document private repositories.

Improvements

- **Optimized AI context processing**: We improved how the system analyzes your codebase, allowing for more accurate documentation generation even for larger, complex repositories.
- **Enhanced synchronization scheduling**: You can now set specific synchronization times based on your local time zone, ensuring your documentation updates exactly when you expect.
- **Style consistency**: We refined the audience presets to ensure that the voice and tone remain consistent across all generated document types.

Fixes

- **Synchronization stability**: We resolved issues where high-volume repositories would occasionally time out during the initial scanning phase.
- **Markdown formatting**: We fixed several inconsistencies in how generated Markdown content was rendered, ensuring cleaner lists and code blocks.
- **URL slug validation**: We corrected errors in the site setup flow that previously allowed invalid characters in URL slugs.

Breaking changes

- **URL slug uniqueness**: URL slugs are now globally unique across the DocuBot platform. If you have an existing slug that conflicts with a new registration, you may be prompted to update your site configuration.
- **Document type mappings**: We revised the default document type mappings for audience presets. When you change your target audiences, your selected document types will reset to the new recommended defaults.

Contact and support

If you have questions about these releases or encounter any issues, please reach out to our support team:

- **Email**: support@ademero.com
- **Support Portal**: https://www.ademero.com
- **Phone**: 863-937-0272

Command reference

Overview

DocuBot operates primarily through a web-based dashboard. You manage your documentation sites, repository sources, and synchronization triggers through the user interface rather than a local command-line environment. The system handles the heavy lifting of code analysis and static site generation on secure, remote workers.

Command Line Interface

A public Command Line Interface (CLI) for external developers or administrators is not available at this time. All configuration, including audience selection, document type refinement, and style application, is performed within the DocuBot workspace.

If you require programmatic access or have specific integration needs, please contact our support team at support@ademero.com or visit https://www.ademero.com.

Manual sync actions

While DocuBot is designed to be automated, you can trigger immediate "command-like" actions from the repository list to refresh your documentation on demand.

Rebuild documentation

This action instructs the DocuBot engine to immediately pull the latest state from your connected GitHub repositories and regenerate the entire documentation set.

- **Usage:** Click the **Rebuild documentation** button on any active repository card.
- **Daily Limit:** Manual syncs are limited to 500 requests per day per repository.
- **Outcome:** The system queues a new sync job. You can monitor the progress (Planning, Writing, Building, and Uploading) directly on the repository card.

Automated sync scheduling

You "command" the system's automated behavior by configuring the sync schedule. These settings determine when the backend engine executes the documentation update sequence.

Parameter	Description
Frequency	Choose between Daily, Weekly, or Monthly updates.
Update Time	The specific time of day (HH:MM) the sync command triggers.
Time Zone	The local time zone used to calculate the update time.
Weekday	For weekly schedules, the specific day the sync runs.
Day of Month	For monthly schedules, the specific calendar day the sync runs.

Support and assistance

If you encounter issues with manual triggers or automated scheduling, you can reach out for help through the following channels:

- **Email:** support@ademero.com
- **Phone:** 863-937-0272
- **Web:** https://www.ademero.com

Status codes

Overview of status codes

DocuBot uses standard HTTP response codes to indicate the success or failure of your requests. Whether you are managing your workspace in the dashboard or accessing your published documentation, these codes provide immediate feedback on the result of your actions.

Success codes (2xx)

Success codes indicate that your request was received, understood, and accepted by the system.

- **200 OK**: The request succeeded. This is the standard response for successful data retrieval, such as loading your repository list or viewing a documentation page.
- **201 Created**: The request succeeded and resulted in the creation of a new resource. You see this after successfully initializing a new documentation site.

Client error codes (4xx)

Client error codes indicate that there was an issue with the request, often due to incorrect configuration or authentication failures.

- **400 Bad Request**: The system cannot process the request due to invalid input. This typically happens if you provide an invalid URL slug or an incorrect configuration payload during setup.
- **401 Unauthorized**: Authentication is required and has failed or was not provided. This occurs if your Google or GitHub session tokens are missing or have expired.
- **403 Forbidden**: You do not have permission to access the resource. This most commonly happens when you try to sync a private repository without granting the necessary GitHub OAuth permissions.
- **404 Not Found**: The requested resource could not be found. You may see this if you navigate to an invalid documentation route or if a repository source is missing.

Server error codes (5xx)

Server error codes indicate that the system encountered an unexpected condition while processing your request.

- **500 Internal Server Error**: A generic error message indicating an unexpected failure during the documentation generation or synchronization process.
- **503 Service Unavailable**: The system is temporarily unable to handle the request. This usually occurs during brief maintenance windows or when the generation engine is experiencing high traffic.

Common causes and resolutions

Use the following table to diagnose and fix the most frequent error states.

Status Code	Common Cause	Resolution
400	Invalid or duplicate URL slug	Ensure your slug uses only lowercase letters, numbers, and dashes, and is unique across DocuBot.
401 / 403	Expired GitHub connection	Navigate to your account settings and re-authenticate your GitHub account to refresh access.
404	Incorrect repository URL	Verify that the GitHub repository URL is correct and that the repository has not been deleted or renamed.
5xx	Generation engine timeout	Wait a few minutes and trigger a manual sync. If the error persists, check your repository for large files that might exceed processing limits.

If you continue to experience issues with specific status codes, contact support at support@ademero.com, visit https://www.ademero.com, or call 863-937-0272.

Contact and support

We’re here to help you get the most out of DocuBot. If you encounter issues or have questions about managing your documentation sites, use the channels below to reach our team.

Support channels

You can reach us through several primary channels for technical assistance and account inquiries.

- **Email**: Send your detailed questions to support@ademero.com.
- **Web**: Visit our support portal and resources at https://www.ademero.com.
- **Phone**: Call us at 863-937-0272 for direct assistance.

Response expectations

Our support team monitors requests during standard business hours. We aim to provide an initial response to all inquiries within one business day. For urgent issues regarding live documentation availability, please use the phone or web portal for the fastest routing.

What to include in your request

To help us resolve your issue efficiently, please include the following information in your support request:

- **Account email**: The email address you use to sign in to the DocuBot dashboard.
- **Issue description**: A clear, detailed explanation of the problem or question.
- **Context**: The specific repository name or URL slug affected by the issue.
- **Steps to reproduce**: A brief list of actions you took before the issue occurred.
- **Error details**: Any specific error messages displayed in the dashboard or sync logs.

Self-service resources

Before reaching out to support, we recommend checking our self-service resources for immediate answers.

- **Searchable help center**: Use the search bar at the top of this site to find specific topics.
- **Troubleshooting guides**: Review the Troubleshooting guide for common sync and configuration fixes.
- **How-to guides**: Consult our Core scenario how-to for step-by-step instructions on standard workflows.