

Cedarville App Portal

First-Time User Guide

Create, publish, manage, and troubleshoot Cedarville apps without prior development or hosting experience.

Includes the complete guide, troubleshooting, FAQ, and glossary.

Cedarville App Portal User Guide

This guide explains the complete app lifecycle in plain language. Start with the [Quick Start](#) if you only need the shortest path to a first published app.

1. What the portal does

The Cedarville App Portal brings the main pieces of an app into one managed workflow. It can create starter code from an approved template, keep that code in a managed GitHub repository, prepare Azure hosting, publish the app, and help you manage access afterward.

The portal does not design every screen or write every business rule for you. After creating the starter, you can work with Codex or a developer to customize the app. GitHub remains the supported source of truth: the version stored there is the version Cedarville tools review and publish.

2. Understand the app lifecycle

- 1 Create or add:** Start from a portal template, add a repository that already exists, or create an empty repository for an app still on your computer.
- 2 Customize:** Use Codex or a developer to change the managed GitHub repository.
- 3 Prepare:** The portal adds and checks the settings needed for GitHub and Azure to work together.
- 4 Publish:** The portal starts the GitHub workflow that sends the current app to Azure.
- 5 Manage:** Use My Apps to check status, change settings, invite collaborators, download a copy, repair setup, or remove resources.

3. Choose the right starting point

Create a new app from a template

Choose **Create App** when you are starting a new project and want Cedarville-approved defaults. Recommended Templates are written for common, non-technical use cases. Developer Starters expose lower-level choices and are better when a developer already knows the intended architecture.

Common template capabilities include:

- A PostgreSQL database for information the app must save.
- Microsoft Entra login when only authorized Cedarville users should enter the app.
- A web interface for forms, trackers, and information pages.
- An API or automation service for a system-to-system process without a normal web page.

Selecting a database or login adds infrastructure and configuration. Choose it because the app needs it, not because it sounds useful.

Add an existing app

Choose **Add Existing App** when code already exists.

- **Already on GitHub:** Give the portal the repository address. If necessary, the portal imports it into Cedarville's managed GitHub organization while preserving its history. The portal checks whether it matches a supported Azure App Service runtime.

- **Not on GitHub yet:** Let the portal create an empty managed repository. The app details page provides instructions that Codex or a developer can use to connect the local folder and push the code.

Imported apps currently support root Next.js, Express, Python FastAPI, and plain static Python apps. A repository with conflicting publishing files may require a GitHub review page before the portal applies its setup.

4. Create an app from a template

- 1 Open **Create App**.
- 2 Read the template summaries and select the closest match.
- 3 Enter a short, recognizable app name. Avoid department abbreviations that coworkers may not understand.
- 4 Describe the app's purpose and intended users.
- 5 Review optional database and login choices.
- 6 Choose **Create Only** if you want to customize before hosting, or **Create and Publish** if the starter is ready for an initial deployment.
- 7 Keep the result page open until repository and publishing setup statuses settle.

The portal also creates a downloadable ZIP. Treat this as a convenience copy. Make ongoing changes in the managed GitHub repository so the portal, Codex, reviewers, and Azure all use the same source.

5. GitHub access and code changes

GitHub is the managed online location for the app's code. It records changes and lets Cedarville's publishing workflow use a reviewed source.

To request access:

- 1 Open **Settings** and save your GitHub username.
- 2 Return to the app from **My Apps**.
- 3 Use the repository access action if access has not already been granted.
- 4 Accept any GitHub invitation sent to your account.

Portal collaboration and GitHub access are related but separate. A portal collaborator can see and operate the app in the portal, but may still need to request GitHub repository access.

When Codex or a developer changes the app, confirm the finished work is committed and pushed to the portal-managed repository. Local files that were never pushed cannot be published by the portal.

6. Publishing to Azure

Publishing makes the current GitHub version available as a running website in Azure.

Before publishing, confirm:

- Repository status is ready.
- Publishing setup is ready.
- Required environment variables have been added.
- The code you want is present in the managed GitHub repository.

Select **Publish** from the app details page. Publishing may take several minutes. You can leave the page and return through My Apps. A successful run shows **Published** and a Published app link.

After publishing, open the app and test its most important task. A successful deployment only proves that Azure started the app; it does not prove every form, permission, integration, or data rule behaves correctly.

Updating an app

- 1 Make and test the change with Codex or a developer.
- 2 Push the change to the managed GitHub repository.
- 3 If push-to-deploy is enabled, the GitHub workflow may publish automatically. Otherwise, return to the portal and select **Publish**.
- 4 Test the published app again.

7. Environment variables and secrets

Environment variables provide settings the app needs at runtime, such as an external service address or secret credential. Add them from the app details page instead of placing secret values in the code or documentation.

The portal stores user-managed secret values in an app-specific Azure Key Vault and gives only that app access. The portal does not show the secret value again after it is saved.

Good practices:

- Use the exact variable name provided by the app or integration instructions.
- Paste only the value, without explanatory text.
- Replace a variable when its credential rotates.
- Republish or restart as directed after changing a required setting.
- Never put real secrets in GitHub files, screenshots, support tickets, or email.

8. Collaborators and ownership

Every app has one primary owner. Owners and administrators can invite Cedarville coworkers by email. The coworker must accept the invitation through Cedarville sign-in before becoming a collaborator.

Collaborators can view app details, download artifacts, request their own GitHub access, repair publishing setup, and publish changes. They cannot delete app resources or transfer ownership. An administrator can reassign the primary owner when responsibility changes.

Removing a collaborator ends portal access immediately. GitHub access is revoked on a best-effort basis when the portal knows the person's GitHub username; verify repository access separately when removing someone from sensitive work.

9. Downloads

The ZIP download is a snapshot of generated starter files and instructions. Owners and collaborators can download it from the app details page. It is useful for review or recovery, but it is not the preferred place for continuing development once a managed repository exists.

If a download returns Not Found, open the app through My Apps and verify that you are the owner or an accepted collaborator. The portal deliberately does not reveal another user's artifact.

10. Repair Publishing Setup

Use **Repair Publishing Setup** when the portal says managed publishing credentials or settings are missing, stale, or out of date. Repair refreshes portal-managed GitHub secrets, Azure connection settings, and federated credentials for that app.

Repair does not delete the repository or Azure app, and it does not start a deployment. When repair finishes, select Publish separately if you want to deploy code.

11. Delete an app carefully

Deletion is divided into separate scopes:

Choice	What it removes
Portal record and artifact	Removes the app from My Apps and deletes its stored ZIP.
GitHub repository	Deletes the managed source repository and its history.
Azure deployment	Deletes the app's Azure Web App and its app-specific database when one exists.

Azure deletion does not remove Cedarville's shared App Service Plan or shared PostgreSQL server. If you delete only the portal record and leave GitHub or Azure unchecked, those resources remain but can no longer be managed from My Apps. Arrange manual cleanup with Cedarville IT if that happens.

Before deleting, confirm the app name, decide whether the code or data must be retained, and coordinate with collaborators. Repository and Azure deletion can be difficult or impossible to reverse.

12. Getting help

Start with [Troubleshooting](#), then review the [FAQ](#). If you still need help, contact Cedarville IT using the approved support channel.

Provide:

- App name.
- Support reference, if shown.
- Approximate time and the action you selected.
- Exact on-screen message.
- A screenshot with sensitive values hidden.

Do not provide passwords, client secrets, database connection strings, environment-variable values, private keys, or downloaded app source unless support specifically arranges a secure transfer.

Troubleshooting

Start with the row that most closely matches what you see. Read the full on-screen message before retrying. Avoid repeatedly selecting an action while it is still running.

What you see	What it usually means	What to do
You cannot sign in	The Cedarville session, account, or portal authentication configuration needs attention.	Close other Microsoft sign-in prompts, use your Cedarville account, and try again. If it continues, contact support.
An app creation form will not submit	A required field is empty or has an unsupported value.	Read the message beside each field, correct it, and submit again.
Repository setup failed	GitHub could not create, import, or prepare the managed repository.	Retry once. Confirm the repository address is correct and accessible. Then provide the displayed support reference to support.
A GitHub invitation is pending	GitHub access was requested but has not been accepted.	Sign in to the correct GitHub account, check notifications and email, and accept the repository invitation.
The portal does not know your GitHub username	Your portal profile is missing the account name used on GitHub.	Open Settings, enter the username only, save it, and return to the app.
Publishing setup says Needs Repair	A managed credential or required setting is stale or missing.	Select Repair Publishing Setup. When setup returns to Ready, select Publish separately.
Apply Publishing Setup is offered	An imported repository still needs the portal's publishing files.	Apply the setup. If the portal reports conflicts, use Review Publishing Changes instead.
Review Publishing Changes is offered	Existing repository files overlap with portal publishing files.	Open the GitHub review page, review and approve the changes, then return to the portal. Ask a developer for help if you do not recognize the files.
Publish failed	GitHub Actions, Azure, the app build, or a required setting failed.	Open the failure details if available. Fix the named issue, retry once, and contact support if it repeats.
Published app shows an error page	Azure started a deployment, but the app may have a code, startup, database, or configuration error.	Confirm required environment variables, then republish. Record the time and error text for support.
Your latest change is missing	The change may not have been pushed to the managed repository or republished.	Confirm the commit is on GitHub, publish again if needed, then refresh the app without using the browser cache.
Download returns Not Found	The artifact is missing or your account does not have access.	Open the app from My Apps. Confirm you are the owner or an accepted collaborator and that a download is listed.
A collaborator cannot open the app in the portal	The invitation may not be accepted or the wrong Cedarville address was used.	Ask the person to use the invitation link and Cedarville sign-in. Remove and resend the invitation if the address is wrong.
A collaborator cannot open GitHub	Portal collaboration does not automatically guarantee GitHub access.	Have the collaborator save their GitHub username and request repository access from the app page.
An environment variable did not fix the app	The name may be wrong, the value may need rotation, or the app may need another publish.	Verify the exact name, replace the value, and follow the app's instructions to republish. Never send the value to support.
A delete option is unavailable	Your role or the current resource state does not allow that deletion.	Confirm you are the primary owner or ask a portal administrator for help.

Safe retry sequence

- 1 Wait for the current action to finish or clearly fail.
- 2 Read and save the exact message and support reference.
- 3 Refresh the page once.
- 4 Use Repair Publishing Setup only when the setup status calls for it.
- 5 Retry the failed action once.
- 6 If the same problem repeats, stop and contact support.

What to send support

- App name and support reference.
- The approximate date and time.
- The page and button you used.
- Exact message or a screenshot with sensitive information hidden.
- Whether the issue happened once or repeated after one retry.

Never send secret values, passwords, private keys, database connection strings, or the contents of environment variables.

Frequently Asked Questions

Do I need to know how to program?

No. Recommended Templates and the portal workflow are designed for first-time users. You may still work with Codex or a developer to customize behavior beyond the starter.

What is GitHub, and why does the portal use it?

GitHub is a managed online place for app code and change history. It gives Codex, reviewers, collaborators, and Azure publishing one supported source instead of passing ZIP files between people.

What is Azure?

Azure is Microsoft's cloud platform. The portal uses Azure App Service to run each published web app. Apps may also receive an app-specific database and secure secret storage when required.

Which template should I choose?

Choose the Recommended Template whose examples most closely resemble your project. Use a Developer Starter only when a developer has identified the needed runtime or service type.

Can I change templates later?

Not as a simple switch. A template shapes the starter code and available features. Ask Codex or a developer whether adapting the current app or creating a new one is safer.

Does publishing make my app public?

Not necessarily. Publishing makes the app run in Azure. Who can enter depends on the app's Microsoft Entra login and authorization rules. Confirm the intended audience before sharing its URL.

What is the difference between Create Only and Create and Publish?

Create Only prepares the app and managed repository without immediately completing the first deployment. Create and Publish also prepares Azure publishing and starts the initial publishing path when setup succeeds.

What happens after someone changes the code?

The change must be committed and pushed to the portal-managed GitHub repository. Then push-to-deploy may publish automatically, or an authorized portal user can select Publish.

Can another person help manage my app?

Yes. The owner or an administrator can invite a Cedarville coworker as a collaborator. The collaborator accepts through Cedarville sign-in and requests separate GitHub access when needed.

What does Repair Publishing Setup do?

It refreshes portal-managed GitHub and Azure publishing credentials and settings. It does not delete resources, change your app's code, or start a deployment.

What is in the downloaded ZIP?

For generated apps, it contains the generated starter snapshot and supporting instructions. It is a convenience and recovery artifact; ongoing development should use the managed GitHub repository.

Where should passwords and API keys go?

Use the app's Environment Variables area when instructed. Do not place real values in code, GitHub files, documentation, screenshots, chat messages, or email.

Can I delete the portal record but keep the live app?

Yes, because deletion scopes are separate. However, if you remove the portal record and keep GitHub or Azure, those resources will no longer appear in My Apps and may require manual IT assistance later.

Can deletion be undone?

Do not assume it can. Deleting a GitHub repository or Azure deployment may permanently remove code, history, configuration, or data. Coordinate and preserve anything required before deleting.

Why does a download say Not Found instead of Access Denied?

The portal deliberately returns a quiet Not Found response for artifacts that do not belong to the signed-in owner or collaborator. This avoids revealing another user's app information.

What should I include in a support request?

Include the app name, support reference, approximate time, action selected, and exact on-screen message. Never include passwords, secret values, private keys, or environment-variable contents.

Glossary

App

A website or service created, tracked, and published through the portal.

App owner

The primary person responsible for an app. Owners can manage collaborators and delete scoped resources.

Azure

Microsoft's cloud platform. The portal uses Azure to host running apps and related app-specific resources.

Codex

An AI coding assistant that can help create or change an app's files. Finished changes must reach the managed GitHub repository before the portal can publish them.

Collaborator

A Cedarville coworker invited to help manage an app in the portal. GitHub repository access is requested separately.

Database

Structured storage used when an app must remember records such as requests, approvals, or tracker entries.

Deployment

A particular attempt to build and send the app to Azure. Publishing starts a deployment.

Environment variable

A named runtime setting supplied outside the app's code. It may contain a normal configuration value or a secret.

GitHub

The service Cedarville uses to store app code, record changes, collaborate, and run publishing workflows.

Managed repository

The private GitHub repository created or imported by the portal and treated as the supported source of truth.

Microsoft Entra login

Cedarville sign-in used to identify users and control who may enter an app.

Publish

Build and send the current managed GitHub version to Azure so it can run in a web browser.

Publishing setup

The GitHub workflow, Azure resources, secrets, and credentials that allow the portal to publish an app safely.

Support reference

A safe identifier that helps Cedarville IT find technical records for a failed or important action.

Template

An approved starting point that supplies common app files, features, and Cedarville defaults.

ZIP artifact

A downloadable snapshot of generated app files. It is useful for convenience or recovery but is not the preferred source for ongoing development.