

# Migrating organizations from GitHub.com to GitHub Enterprise Cloud

## In this article

About organization migrations with GitHub Enterprise Importer

Prerequisites

Step 0: Get ready to use the GitHub GraphQL API

Step 1: Get the enterprise ID for your migration destination

Step 2: Start your organization migration

Step 3: Check the status of your migration

Step 1: Install the GEI extension of the GitHub CLI

Step 2: Update the GEI extension of the GitHub CLI

Step 3: Set environment variables

Step 4: Migrate your organization

Step 5: Validate your migration and check the error log

Step 4: Validate your migration and check the error log

---

You can migrate organizations from GitHub.com to GitHub Enterprise Cloud, using the GitHub CLI or the GraphQL API.

API   GitHub CLI

---

## About organization migrations with GitHub Enterprise Importer

---

You can run your migration with either the GitHub CLI or the API.

The GitHub CLI simplifies the migration process and is recommended for most customers. Advanced customers with heavy customization needs can use the API to build their own integrations with GitHub Enterprise Importer.

To see instructions for using the API, use the tool switcher at the top of the page.

To see instructions for using the GitHub CLI, use the tool switcher at the top of the page.

## Prerequisites

---

- To ensure you understand the known support limitations of the Importer, review "[About GitHub Enterprise Importer](#)."
- We strongly recommend that you perform a trial run of your migration and complete your production migration soon after. To learn more about trial run best practices, see "[Preparing to run a migration with GitHub Enterprise Importer](#)."
- While not required, we recommend halting work during your production migration. The Importer doesn't support delta migrations, so any changes that happen during the migration will not migrate. If you choose not to halt work during your production migration, you'll need to manually migrate these changes.
- For the source organization, you must be an organization owner or have the migrator

role. For more information, see "[Granting the migrator role for GitHub Enterprise Importer](#)."

- For the destination enterprise account, you must be an enterprise owner.

## Step 0: Get ready to use the GitHub GraphQL API

To make GraphQL queries, you'll need to write your own scripts or use an HTTP client like [Insomnia](#).

To learn more about getting started with the GitHub GraphQL API, including how to authenticate, see "[Forming calls with GraphQL](#)."

## Step 1: Get the enterprise ID for your migration destination

As an enterprise owner in GitHub.com, use the following query to return the ID for the enterprise account you want to own the migrated organization. You'll need the enterprise ID to identify your migration destination.

```
query(  
  $slug: String!  
) {  
  enterprise(slug: $slug)  
  {  
    slug  
    id  
  }  
}
```

### Query variable

### Description

slug

The slug for your enterprise account, which you can identify by looking at the URL for your enterprise,  
`https://github.com/enterprises/SLUG`.

## Step 2: Start your organization migration

When you start a migration, a single organization and its accompanying data migrates into a brand new organization within the destination enterprise that you identify.

```
mutation startOrganizationMigration (  
  $sourceOrgUrl: URI!,  
  $targetOrgName: String!,  
  $targetEnterpriseId: ID!,  
  $sourceAccessToken: String!,  
  $targetAccessToken: String!  
) {  
  startOrganizationMigration(input: {  
    sourceOrgUrl: $sourceOrgUrl,  
    targetOrgName: $targetOrgName,  
    targetEnterpriseId: $targetEnterpriseId,  
    sourceAccessToken: $sourceAccessToken,  
    targetAccessToken: $targetAccessToken  
  }) {  
    orgMigration {  
      id  
    }  
  }  
}
```

```
}
```

Query variable	Description
<code>sourceOrgUrl</code>	The URL of the source organization, such as <code>https://github.com/octo-org</code> .
<code>targetOrgName</code>	The name you want the new organization to have. Must be unique on GitHub.com.
<code>targetEnterpriseId</code>	The ID of the enterprise that you want to create the new organization in, returned by step 2.
<code>sourceAccessToken</code>	Your personal access token (classic) for the source organization. For requirements, see " <a href="#">Managing access for GitHub Enterprise Importer</a> ."
<code>targetAccessToken</code>	Your personal access token (classic) for the destination enterprise.

In the next step, you'll use the migration ID returned from the `startOrganizationMigration` mutation to check the migration status.

## Step 3: Check the status of your migration

To detect any migration failures and ensure your migration is working, you can query the `OrganizationMigration` (s) that you have created to see the migration status using the `getMigration` query.

The query will return with a status to let you know if the migration is `queued` , `in progress` , `failed` , or `completed` , plus information about how many repositories are waiting to be migrated. If your migration failed, the Importer will provide a reason for the failure.

```
query (
  $id: ID!
){
  node( id: $id ) {
    ... on OrganizationMigration {
      id
      sourceOrgUrl
      targetOrgName
      state
      failure_reason
      remaining_repositories_count
      total_repositories_count
    }
  }
}
```

Query variable	Description
<code>id</code>	The <code>id</code> of your migration.

## Step 1: Install the GEI extension of the GitHub CLI

If this is your first migration, you'll need to install the GEI extension of the GitHub CLI. For more information about the GitHub CLI, see "[About GitHub CLI](#)."

- 1 Install the GitHub CLI. For installation instructions for GitHub CLI, see the [GitHub CLI repository](#).

**Note:** You need version 2.4.0 or newer of GitHub CLI. You can check the version you have installed with the `gh --version` command.

- 2 Install the GEI extension.

Shell



```
gh extension install github/gh-gei
```

Any time you need help with the GEI extension, you can use the `--help` flag with a command. For example, `gh gei --help` will list all the available commands, and `gh gei migrate-repo --help` will list all the options available for the `migrate-repo` command.

## Step 2: Update the GEI extension of the GitHub CLI [↗](#)

The GEI extension is updated weekly. To make sure you're using the latest version, update the extension.

```
gh extension upgrade github/gh-gei
```

## Step 3: Set environment variables [↗](#)

Before you can use the GEI extension to migrate to GitHub Enterprise Cloud, you must create personal access tokens (classic) that can access the source organization and destination enterprise, then set the personal access tokens (classic) as environment variables.

- 1 Create and record a personal access token that meets all the requirements to authenticate for the source organization for organization migrations. For more information, see "[Managing access for GitHub Enterprise Importer](#)."
- 2 Create and record a personal access token (classic) that meets all the requirements to authenticate for the destination enterprise for organization migrations.
- 3 Set environment variables for the personal access tokens (classic), replacing TOKEN in the commands below with the personal access tokens (classic) you recorded above. Use `GH_PAT` for the destination enterprise and `GH_SOURCE_PAT` for the source organization.
  - If you're using Terminal, use the `export` command.

Shell



```
export GH_PAT="TOKEN"
export GH_SOURCE_PAT="TOKEN"
```

- If you're using PowerShell, use the `$env` command.


Shell



```
$env:GH_PAT="TOKEN"
$env:GH_SOURCE_PAT="TOKEN"
```

## Step 4: Migrate your organization

To migrate an organization, use the `gh gei migrate-org` command.

Shell 

```
gh gei migrate-org --github-source-org SOURCE --github-target-org DESTINATION --
github-target-enterprise ENTERPRISE
```

Replace the placeholders in the command above with the following values.

Placeholder	Value
SOURCE	Name of the source organization
DESTINATION	The name you want the new organization to have. Must be unique on GitHub.com.
ENTERPRISE	The slug for your destination enterprise, which you can identify by looking at the URL for your enterprise account, <code>https://github.com/enterprises/SLUG</code> .

## Step 5: Validate your migration and check the error log

## Step 4: Validate your migration and check the error log

After your migration has finished, we recommend that you check the migration log repository. For more information, see "[Accessing your migration logs for GitHub Enterprise Importer](#)."

Finally, we recommend you perform a soundness check of your organization and migrated repositories.

### Legal