

Granting the migrator role for GitHub Enterprise Importer

In this article

About the migrator role

Granting the migrator role with the ADO2GH extension

Granting the migrator role with the BBS2GH extension

Granting the migrator role with the GEI extension

Granting the migrator role with the GraphQL API

To allow someone who isn't an organization owner to run migrations, you can grant a person or team the migrator role.

Who can use this feature

Organization owners can grant the migrator role.

API GitHub CLI

About the migrator role [↗](#)

To remove the need for organization owners to complete migrations, GitHub.com includes a distinct role for using GitHub Enterprise Importer. Granting the migrator role allows you to designate other teams or individuals to handle your migrations.

You can grant the migrator role to an individual user or a team. We strongly recommend that you assign the migrator role to a team. Then, you can further customize who can run a migration by adjusting team membership. For more information about changing team membership, see "[Adding organization members to a team](#)" or "[Removing organization members from a team](#)."

You can only grant the migrator role for an organization on GitHub.com. If you're migrating a repository between two organizations on GitHub.com, you can grant the migrator role to the same person or team for both organizations, but you must grant each separately.


Note: You cannot grant the migrator role for enterprise accounts. Therefore, you can only run an organization migration if you're an owner of the destination enterprise. However, you can grant the migrator role to that enterprise owner for the source organization.

After you grant the migrator role, make sure the migrator uses a personal access token that meets all the requirements for running migrations. For the details of those requirements, see "[Managing access for GitHub Enterprise Importer](#)."


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

Granting the migrator role with the ADO2GH extension [↗](#)


- 1 On GitHub.com, create and record a personal access token that meets all the requirements for granting the migrator role. For a full list of requirements, see "[Managing access for GitHub Enterprise Importer](#)."
- 2 Set the personal access token as an environment variable, replacing TOKEN in the commands below with the personal access token you recorded above.
 - If you're using Terminal, use the `export` command.

```
Shell   
  
export GH_PAT="TOKEN"
```

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


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

- 3 Use the `gh ado2gh grant-migrator-role` command, replacing ORGANIZATION with the organization you want to grant the migrator role for, ACTOR with the user or team name, and TYPE with `USER` or `TEAM`.


```
Shell   
  
gh ado2gh grant-migrator-role --github-org ORGANIZATION --actor ACTOR --  
actor-type TYPE
```

Granting the migrator role with the BBS2GH extension

- 1 On GitHub.com, create and record a personal access token that meets all the requirements for granting the migrator role. For a full list of requirements, see "[Managing access for GitHub Enterprise Importer](#)."
- 2 Set the personal access token as an environment variable, replacing TOKEN in the commands below with the personal access token you recorded above.
 - If you're using Terminal, use the `export` command.

```
Shell   
  
export GH_PAT="TOKEN"
```

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

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

- 3 Use the `gh bbs2gh grant-migrator-role` command, replacing ORGANIZATION with

the organization you want to grant the migrator role for, ACTOR with the user or team name, and TYPE with `USER` or `TEAM`.

Shell



```
gh bbs2gh grant-migrator-role --github-org ORGANIZATION --actor ACTOR --actor-type TYPE
```

Granting the migrator role with the GEI extension [↗](#)

Note: The GitHub CLI does not support granting the migrator role for organizations on GitHub Enterprise Server, so you must be an organization owner of the source organization to migrate repositories from GitHub Enterprise Server.

- 1 On GitHub.com, create and record a personal access token that meets all the requirements for granting the migrator role. For a full list of requirements, see "[Managing access for GitHub Enterprise Importer](#)."
- 2 Set the personal access token as an environment variable, replacing TOKEN in the commands below with the personal access token you recorded above.
 - If you're using Terminal, use the `export` command.

Shell



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

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

Shell



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

- 3 Use the `gh gei grant-migrator-role` command, replacing ORGANIZATION with the organization you want to grant the migrator role for, ACTOR with the user or team name, and TYPE with `USER` or `TEAM`.

Shell



```
gh gei grant-migrator-role --github-org ORGANIZATION --actor ACTOR --actor-type TYPE
```

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

Granting the migrator role with the GraphQL API [↗](#)

You can use the `grantMigratorRole` GraphQL mutation to assign the migrator role and the `revokeMigratorRole` mutation to revoke the migrator role.

You must use a personal access token (PAT) that meets all access requirements. For

more information, see "[Managing access for GitHub Enterprise Importer](#)."

grantMigratorRole mutation

This GraphQL mutation sets the migration role.

```
mutation grantMigratorRole (
  $organizationId: ID!,
  $actor: String!,
  $actor_type: ActorType!
) {
  grantMigratorRole( input: {
    organizationId: $organizationId,
    actor: $actor,
    actorType: $actor_type
  })
  { success }
}
```

Query variable	Description
organizationId	The <code>ownerId</code> (or organization ID) for your organization, from the <code>GetOrgInfo</code> query.
actor	The team or username who you want to assign the migration role to.
actor_type	Specify whether the migrator is a <code>USER</code> or <code>TEAM</code> .

revokeMigratorRole mutation

This mutation removes the migrator role.

```
mutation revokeMigratorRole (
  $organizationId: ID!,
  $actor: String!,
  $actor_type: ActorType!
) {
  revokeMigratorRole( input: {
    organizationId: $organizationId,
    actor: $actor,
    actorType: $actor_type
  })
  { success }
}
```

Legal