



Using the CLI

The Railway Command Line Interface (CLI) lets you interact with your Railway project from the command line.



Installing the CLI

The Railway CLI can be installed via Homebrew, npm, Scoop, or directly from the source.

Homebrew (macOS)

In a Terminal, enter the following command:

```
brew install railway
```

npm (macOS, Linux, Windows)

In a Terminal, enter the following command:

```
npm i -g @railway/cli
```

This requires version \Rightarrow 16 of Node.js.

Shell Script (macOS, Linux, Windows via WSL)

In a Terminal, enter the following command:

```
bash <(curl -fsSL cli.new)
```

On Windows, you should use Windows Subsystem for Linux with a Bash shell.

Scoop (Windows)

In a PowerShell terminal, enter the following command:

```
scoop install railway
```

This installs a native Windows binary (`.exe`). To learn more about Scoop, see <https://scoop.sh/>.

Pre-built Binaries

We publish pre-built binaries on our GitHub repository that you can download and use directly.

From Source

The Railway CLI is an open source project on GitHub. You can build a binary from source if you wish.

Authenticating with the CLI

Before you can use the Railway CLI, you must authenticate the CLI to your Railway account:

```
railway login
```

This command opens a new tab in your default browser to the <https://railway.app> authentication page. Follow the instructions to complete the authentication process.

Manual Login

You can also authenticate manually using a Pairing Code. This can be useful if you're authenticating the CLI inside an environment without a browser (e.g. SSH sessions).

Use the `--browserless` flag to authenticate manually:

```
railway login --browserless
```

Tokens

In situations where user input or interaction isn't possible, such as in CI/CD pipelines, you can set either the `RAILWAY_TOKEN` or `RAILWAY_API_TOKEN` environment variable, based on your specific requirements as detailed below.

A Project Token is set via the `RAILWAY_TOKEN` environment variable.

An Account or Team Token is set via the `RAILWAY_API_TOKEN` environment variable.

Note: You can only use one type of token at a time. If both are set, the `RAILWAY_TOKEN` variable will take precedence.

Project Tokens

You can use Project Tokens to authenticate project-level actions.

Project Tokens allow the CLI to access all the project-level actions in the environment set when the token was created.

Some actions you can perform with a project token include -

- Deploying code - `railway up`
- Redeploying a deployment - `railway redeploy`
- Viewing build and deployment logs - `railway logs`

Some actions you **cannot** perform with a project token include -

- Creating a new project - `railway init`
- Printing information about the user - `railway whoami`

Use the token by setting the `RAILWAY_TOKEN` environment variable and then running `railway <command>` .

```
RAILWAY_TOKEN=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX railway up
```

Account Tokens

Account Tokens come in two types - Personal Account Tokens and Team Tokens.

You can use Account Tokens to authenticate all CLI actions across all workspaces.

However, you can only use Team tokens to authenticate actions on projects within the workspace the token was scoped to when it was created.

Some actions you can perform with a personal account token include -

- Creating a new project - `railway init`
- Printing information about the user - `railway whoami`

Some actions you **cannot** perform with Team Token include -

- Printing information about the user - `railway whoami`
- Linking to another workspace - `railway link`

Use the token by setting the `RAILWAY_API_TOKEN` environment variable and then running `railway <command>` .

```
RAILWAY_API_TOKEN=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX railway whoami
```

Common Examples of CLI Usage

Below are some of the most commonly used CLI commands. Find a complete list of CLI commands in the [CLI API reference page](#).

Link to a Project

To associate a project and environment with your current directory:

```
# Link to a project  
railway link
```

This prompts you to select a team, project, and environment to associate with your current directory. Any future commands will be run against this project and environment.

Link to a Service

Associate a service in a project and environment with your current directory:

```
# Link to a service  
railway service
```

This links your current directory with the chosen service.

Create a Project

Create a new project directly from the command line.

```
# Create a new project  
railway init
```

This prompts you to name your project and select a team to create the project in.

Local Development

Run code locally with the same environment variables as your Railway project.

```
# Run <cmd> locally with the same environment variables as your Railway project  
railway run <cmd>
```

For example, to run your Node.js project with your remote environment variables:

```
# Run your Node.js project with your remote environment variables  
railway run npm start
```

Local Shell

Open a new local shell with Railway environment variables. Similar to `railway run` but opens a new shell.

```
# Open a new shell with Railway environment variables  
railway shell
```

Environments

Projects might have multiple environments, but by default the CLI links to the `production` environment. You can change the linked environment with the `environment` command.

```
# Change the linked environment  
railway environment
```

Deploy

Deploy the linked project directory (if running from a subdirectory, the project root is still deployed).

```
# Show build logs
railway up

# Return immediately after uploading
railway up --detach
```

If there are multiple services within your project, the CLI will prompt you for a service to deploy to.

Add Database Service

Provision database services for a project.

```
railway add
```

Prompts you to select one or more databases to provision for your project.

Logout

```
railway logout
```

Contributing

Our CLI is open source. Contribute to the development of the Railway CLI by opening an issue or Pull Request on our [GitHub Repo](#).

[You can see the full documentation of the CLI API here.](#)



Not finding the answer you need?

Start a Discussion

Environments

Join Priority Boarding!

[Edit this file on GitHub](#)