

Introduction

Set up

Prepare the app

Deploy the app

View logs

Define a Procfile

Scale the app

Declare app dependencies

Run the app locally

Push local changes

Provision add-ons

Start a one-off dyno

Define config vars

Use a database

Next steps

Define config vars

Heroku lets you externalize configuration - storing data such as encryption keys or external resource addresses in config vars.

At runtime, config vars are exposed as environment variables to the application. For example, modify `app/views/welcome/index.erb` so that the method repeats an action depending on the value of the `TIMES` environment variable.

```
<h1>Getting Started with Ruby</h1>

<p>
  Welcome!
</p>

<% for i in 0..(ENV['TIMES'] ? ENV['TIMES'].to_i : 2) do %>
  <p>Hello World #<%= i %>!</p>
<% end %>
```

`heroku local` will automatically set up the environment based on the contents of the `.env` file in your local directory. In the top-level directory of your project there is already a `.env` file that has the following contents:

```
TIMES=10
```

If you run the app with `heroku local`, you'll see "Hello World" ten times.

To set the config var on Heroku, type the following:

```
> heroku config:set TIMES=10
```

View the config vars that are set using `heroku config :`

```
> heroku config
== polar-inlet-4930 Config Vars
PAPERTRAIL_API_TOKEN: erdKhPeeeehIcdFY7ne
TIMES: 10
```

Deploy your changed application to Heroku to see this in action.

Log in to report a problem (/login?back_to=%2Farticles%2Fgetting-started-with-jruby%23define-config-vars)

I understand config vars (define-config-vars)

(Log in (/login?back_to=%2Farticles%2Fgetting-started-with-jruby%23define-config-vars) to save and track your progress)