

[Ruby\\_](https://cloud.google.com/ruby/) (<https://cloud.google.com/ruby/>) [Guides](#)

# Logging app events with Ruby

This part of the Bookshelf app tutorial shows how an app can incorporate detailed logging to help with detecting, debugging, and monitoring potential issues. Logging app events can help you identify issues and solve problems, both during development and after your app is in production.

This page is part of a multipage tutorial. To start from the beginning and read the setup instructions, go to [Ruby Bookshelf app](https://cloud.google.com/ruby/getting-started/tutorial-app) (<https://cloud.google.com/ruby/getting-started/tutorial-app>).

## Installing dependencies

Go to the `getting-started-ruby/5-logging` directory, and enter the following command:

```
bundle install
```



## Configuring settings

1. Copy the example settings file.

```
cp config/settings.example.yml config/settings.yml
```



2. Edit the `settings.yml` file. the same way you did in the [Authenticating users](https://cloud.google.com/ruby/getting-started/authenticate-users) (<https://cloud.google.com/ruby/getting-started/authenticate-users>) section.

For example, suppose your web app client ID is `XYZCLIENTID`, your client secret is `XYZCLIENTSECRET`, your project name is `my-project`, and your Cloud Storage bucket name is `my-bucket`. Then the default section of your `settings.yml` file would look like this:

```
default: &default
  project_id: my-project
  gcs_bucket: my-bucket
  oauth2:
    client_id: XYZCLIENTID
    client_secret: XYZCLIENTSECRET
```



### 3. Copy the example database file.

```
cp config/database.example.yml config/database.yml
```



### 4. Configure the sample app to use the same database that you set up during the [Using structured data](https://cloud.google.com/ruby/getting-started/using-structured-data) (https://cloud.google.com/ruby/getting-started/using-structured-data) portion of this tutorial.

CLOUD SQL

POSTGRES SQL

DATASTORE

- Edit `database.yml`. Uncomment the lines in the Cloud SQL portion of the file.

```
mysql_settings: &mysql_settings
  adapter: mysql2
  encoding: utf8
  pool: 5
  timeout: 5000
  username: [MYSQL_USER]
  password: [MYSQL_PASS]
  database: [MYSQL_DATABASE]
  socket: /cloudsql/[YOUR_INSTANCE_CONNECTION_NAME]
```



- Replace `[MYSQL_USER]` and `[MYSQL_PASS]` with your Cloud SQL instance username and password that you created previously.
- Replace `[MYSQL_DATABASE]` with the name of the database that you created previously.
- Replace `[YOUR_INSTANCE_CONNECTION_NAME]` with the **Instance Connection Name** of your Cloud SQL instance.

**Note:** You can retrieve the Cloud SQL instance connection name by running `gcloud beta sql instances describe [YOUR_INSTANCE_NAME]`.

- Run migrations.

```
bundle exec rake db:migrate
```



## Running the app on your local machine

1. Start a local web server.

```
bundle exec rails server
```



2. In your web browser, enter the following address:

<http://localhost:3000> (`http://localhost:3000`)

To exit the local web server, press `Control+C`.

## Deploying the app to the App Engine flexible environment

1. Compile the JavaScript assets for production.

```
RAILS_ENV=production bundle exec rake assets:precompile
```



2. Deploy the sample app.

```
gcloud app deploy
```



3. In your web browser, enter the following address.

```
https://[YOUR_PROJECT_ID].appspot.com
```



If you update your app, you can deploy the updated version by entering the same command you used to deploy the app the first time. The new deployment creates a new version (<https://console.cloud.google.com/appengine/versions>) of your app and promotes it to the default version. The older versions of your app remain, as do their associated VM instances. Be aware that all of these app versions and VM instances are billable resources.

You can reduce costs by deleting the non-default versions of your app.

To delete an app version:


1. In the Cloud Console, go to the **Versions** page for App Engine.

**GO TO THE VERSIONS PAGE** ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/APPENGINE/VERSIONS](https://console.cloud.google.com/appengine/versions))

2. Select the checkbox for the non-default app version you want to delete.

**Note:** The only way you can delete the default version of your App Engine app is by deleting your project. However, you can [stop the default version in the Cloud Console](https://console.cloud.google.com/appengine/versions) (<https://console.cloud.google.com/appengine/versions>). This action shuts down all instances associated with the version. You can restart these instances later if needed.

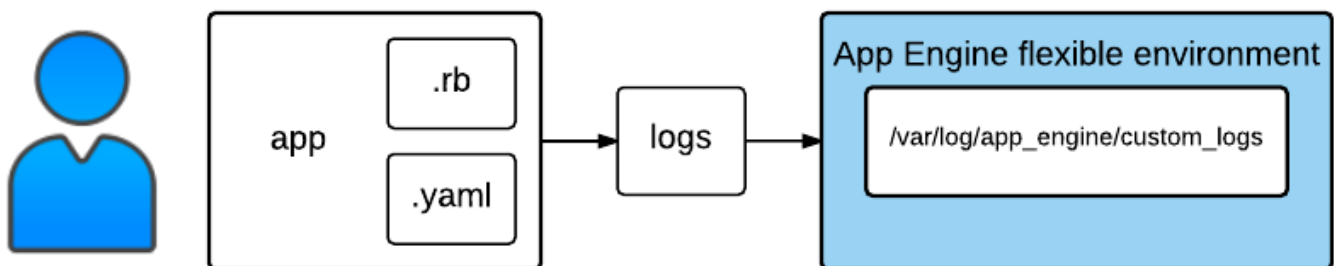
In the App Engine standard environment, you can stop the default version only if your app has manual or basic scaling.

3. Click **Delete**  to delete the app version.

For complete information about cleaning up billable resources, see the [Cleaning up](https://cloud.google.com/ruby/getting-started/using-pub-sub#clean-up) (<https://cloud.google.com/ruby/getting-started/using-pub-sub#clean-up>) section in the final step of this tutorial.

## App structure

The following diagram shows how the app handles logging when deployed to the App Engine flexible environment.



## Viewing logs

As the bookshelf app runs, it writes logging data that is collected and made available in the Google Cloud Console. You can use the [log monitoring tools](#) (<https://console.cloud.google.com/logs>) in the Cloud Console to analyze the logs directly. If you want more detailed analysis, you can use the Cloud Console to stream or import the app's logs into [BigQuery](#) ([https://cloud.google.com/logging/docs/install/logs\\_export](https://cloud.google.com/logging/docs/install/logs_export)) or export them to a [Cloud Storage bucket](#) ([https://cloud.google.com/logging/docs/install/logs\\_export](https://cloud.google.com/logging/docs/install/logs_export)).

The screenshot shows the Google Cloud Console Logs interface. At the top, there are tabs for 'Logs' and 'Exports'. Below that is a search bar labeled 'Filter by label or text search'. There are several dropdown menus for filtering: 'App Engine', 'Default Module', '1 (default)', 'Request', and 'Any log level'. To the right, there is a date range selector 'Up to: Apr 24, 2015, 3:49:40 PM PDT' and buttons for play and refresh. The main area displays a table of log entries for the date 2015-04-24, scanned from 2015-04-15 (17:11:45) to 2015-04-22 (13:26:11). The table has columns for time, status code, response size, response time, and request path. The log entries are as follows:

Time	Status	Response Size	Response Time	Request Path
10:22:10.409	302	288 B	228ms	/books/add
10:22:10.629	200	1.54 KB	175ms	/books/6278949055234048
10:22:12.968	200	1.54 KB	233ms	/books/6278949055234048
10:22:15.391	200	2.91 KB	230ms	/books/6278949055234048
10:22:18.213	200	3.66 KB	304ms	/books
11:04:16.979	200	0 B	3,540.46s	/_ah/background
12:03:17.987	200	0 B	3,541.02s	/_ah/background

## Understanding the code

Rails includes a powerful and configurable logging system. The Bookshelf app configures Rails to place logs in the directory where the App Engine flexible environment expects to find them.

### [5-logging/config/environments/production.rb](#)

(<https://github.com/GoogleCloudPlatform/getting-started-ruby/blob/steps/5-logging/config/environments/production.rb>)

TFORM/GETTING-STARTED-RUBY/BLOB/STEPS/5-LOGGING/CONFIG/ENVIRONMENTS/PRODUCTION.RB)

```
if Dir.exist? "/var/log/app_engine/custom_logs"
  config.logger = ActiveSupport::TaggedLogging.new Logger.new("/var/log/app_engine/c
end
```

In addition to the default messages provided by Rails, you can log anything you want by using the Rails logger object.

```
logger.debug "Debug message."
logger.info "Informational."
logger.fatal "Fatal error."
```

[← PREV \(HTTPS://CLOUD.GOOGLE.COM/RUBY/GETTING-STARTED/AUTHENTICATE-USERS\)](https://cloud.google.com/ruby/getting-started/authenticate-users)

[NEXT > \(HTTPS://CLOUD.GOOGLE.COM/RUBY/GETTING-STARTED/USING-PUB-SUB\)](https://cloud.google.com/ruby/getting-started/using-pub-sub)

---

*Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (https://developers.google.com/terms/site-policies). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated November 15, 2019.*